

Error Control and Concealment for Video Communication: A Review

YAO WANG, MEMBER, IEEE, AND QIN-FAN ZHU, MEMBER, IEEE

The problem of error control and concealment in video communication is becoming increasingly important because of the growing interest in video delivery over unreliable channels such as wireless networks and the Internet. This paper reviews the techniques that have been developed for error control and concealment in the past 10–15 years. These techniques are described in three categories according to the roles that the encoder and decoder play in the underlying approaches. Forward error concealment includes methods that add redundancy at the source end to enhance error resilience of the coded bit streams. Error concealment by postprocessing refers to operations at the decoder to recover the damaged areas based on characteristics of image and video signals. Last, interactive error concealment covers techniques that are dependent on a dialogue between the source and destination. Both current research activities and practice in international standards are covered.

Keywords—Error concealment, error control in video transport, video communications.

I. INTRODUCTION

One inherent problem with any communications system is that information may be altered or lost during transmission due to channel noise. The effect of such information loss can be devastating for the transport of compressed video because any damage to the compressed bit stream may lead to objectionable visual distortion at the decoder. In addition, real-time/interactivity requirements exclude the deployment of some well-known error-recovery techniques for certain applications. Last, issues such as audio-visual synchronization and multipoint communications further complicate the problem of error recovery.

Transmission errors can be roughly classified into two categories: *random bit errors* and *erasure errors*. Random bit errors are caused by the imperfections of physical channels, which result in bit inversion, bit insertion, and bit deletion. Depending on the coding methods and the affected information content, the impact of random bit errors can range from negligible to objectionable. When fixed-length

coding is used, a random bit error will only affect one code word, and the caused damage is generally acceptable. But if variable length coding (VLC) (for example, Huffman coding) is used, random bit errors can desynchronize the coded information such that many following bits are undecodable until the next synchronization code word appears. In some cases, even after synchronization is obtained, decoded information can still be useless since there is no way to determine which spatial or temporal locations correspond to the decoded information. Erasure errors, on the other hand, can be caused by packet loss in packet networks, burst errors in storage media due to physical defects, or system failures for a short time. Random bit errors in VLC can also cause effective erasure errors since a single bit error can lead to many following bits' being undecodable and hence useless. The effect of erasure errors (including those due to random bit errors) is much more destructive than random bit errors due to the loss or damage of a contiguous segment of bits. Since almost all the state-of-the-art video-compression techniques use VLC in one way or another, there is no need to treat random bit errors and erasure errors separately. The generic term "transmission errors" will be used throughout this paper to refer to both random bit errors and erasure errors.

To illustrate the visual artifacts caused by transmission errors, Fig. 1 shows two reconstructed video frames from a MPEG-2¹ coded video sequence when it is delivered over a wireless asynchronous transfer mode (ATM) network. The video is coded at 6 Mbps, and the cell loss rate of the network is 10^{-3} . In this example, the video sequence is divided into groups of pictures (GOP's), with each GOP consisting of 15 frames. The first frame in each GOP is coded in the intramode, referred to as an I-frame, while the remaining frames are coded in the forward interframe prediction mode, called P-frames. Each frame is partitioned into slices, with each slice containing all the 16×16 macroblocks in the same row. A start code is inserted at the beginning of each slice so that the error in a slice will not affect the decoding of the next slice. Any loss in the middle of a slice will render the remaining

Manuscript received July 31, 1997; revised December 5, 1997. The Guest Editor coordinating the review of this paper and approving it for publication was T. Chen.

Y. Wang is with the Department of Electrical Engineering, Polytechnic University, Brooklyn, NY 11201 USA.

Q.-F. Zhu is with PictureTel Corp., Andover, MA 01810 USA.

Publisher Item Identifier S 0018-9219(98)03516-6.

¹MPEG-2 is a standard of the Motion Pictures Experts Group (MPEG) of the International Standards Organization (ISO).



(a)



(b)

Fig. 1. Two reconstructed frames from an MPEG-2 coded sequence when it is delivered over a wireless ATM network. The video is coded at 6 Mbps. The cell-loss rate of the network is 10^{-3} . All the macroblocks in a slice following a damaged macroblock are replaced by the corresponding macroblocks in the previous frame. (a) Frame 15, an I-frame, which has three cell losses. The first two cell losses occur in two consecutive slices in the upper part of the tree trunk. The third cell loss happens in the lower part of the tree trunk. (b) Frame 22, a P-frame, which has a single cell loss at a lower part of the tree, above the location of the third cell loss in Frame 15. One can observe the effect of the cell loss of this frame as well as those propagated from Frame 15.

blocks in this slice undecodable. Furthermore, the damaged blocks in an I-frame will cause reconstruction errors in the following P-frames. In this example, a damaged macroblock is simply replaced by the corresponding macroblock in the reconstructed previous frame, which causes a visible discontinuity when the damaged block falls in a region with fast motion. The first image shown in Fig. 1 is an I-

frame, where three cell losses lead to three damaged slices. The second image is a P-frame, which has a single cell loss. Visible distortions appear in more than one slice, however, because of the error propagation effect. When the reconstructed video is played back in real time, these distortions are visually annoying and are certainly not acceptable for entertainment applications.

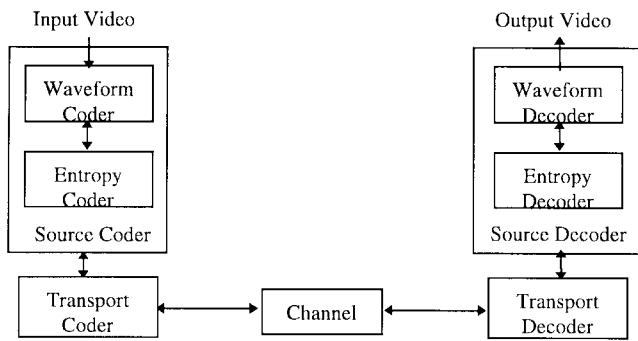


Fig. 2. A functional block diagram for a video communications system.

Techniques for combating transmission errors for video communication have been developed along two avenues. On one hand, traditional error control and recovery schemes for data communications have been extended for video transmission. These techniques aim at lossless recovery. Examples of such schemes include forward error correction (FEC) or, more generally, error control coding (ECC), and automatic retransmission request (ARQ). On the other hand, signal-reconstruction and error-concealment techniques have been proposed that strive to obtain a close approximation of the original signal or attempt to make the output signal at the decoder least objectionable to human eyes. Note that unlike data transmission, where lossless delivery is required absolutely, human eyes can tolerate a certain degree of distortion in image and video signals. In this paper, we attempt to summarize and critique the approaches that have been developed for error control and concealment in the past 10–15 years. The rest of this paper is organized as follows. Section II describes the various components involved in a visual communications system and categorizes the approaches to the error control and concealment problem. Section III reviews techniques for error detection. Sections IV–VI present the error-concealment methods in different categories. Last, Section VII draws some concluding remarks.

II. PROBLEM FORMULATION AND CATEGORIZATION OF APPROACHES

Fig. 2 shows a functional block diagram of a real-time video communications system. The input video is compressed by the source encoder to the desired bit rate. The *transport coder* in the figure refers to an ensemble of devices performing channel coding, packetization and/or modulation, and transport-level control using a particular transport protocol. This transport coder is used to convert the bit-stream output from the source coder into data units suitable for transmission. At the receiver side, the inverse operations are performed to obtain the reconstructed video signal for display. Note that although we only show a one-way transmission, we use double arrows to emphasize the fact that for some applications, there is a backward channel to convey information from the decoder to the encoder side for system control and error concealment.

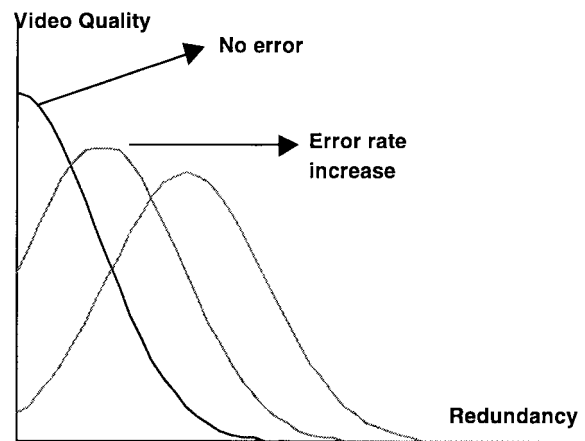


Fig. 3. Illustration of the relationship between the quality of reconstructed video at the decoder and the amount of concealment redundancy employed under different channel error rates. The total bandwidth used for source and channel coding is fixed.

The *source coder* can be further partitioned into two components: the *waveform coder* and the *entropy coder*. The waveform coder is a lossy device that reduces the bit rate by representing the original video using some transformed variables and applying quantization. Examples of waveform coders include transform coding using the discrete cosine transform (DCT) and wavelet transforms, as well as vector quantization. The entropy coder, on the other hand, is a lossless device that maps the output symbols from the waveform coder into binary code words according to the statistical distribution of the symbols to be coded. Examples of entropy-coding methods include Huffman coding and arithmetic coding. Although the waveform coder can use any known video-coding method, we will mainly focus on the type of hybrid coder that uses DCT and motion-compensated prediction. This coding scheme has been proven to be the most effective for a broad range of applications and is the basis for all current video-coding standards [1]–[3]. The transport coder can vary for different applications. Examples of real-time transport protocols include H.221 in H.320, H.223 in H.324, and H.225 in H.323 [4]–[9].

In general, to help error detection and concealment at the decoder, a certain amount of redundancy needs to be added at the waveform-, entropy-, or transport-coder level. We refer to such added redundancy as *concealment redundancy*. Fig. 3 illustrates qualitatively the dependency of the reconstructed video quality on the concealment redundancy and channel error rate. Here, we assume that the total bit rate used for source and channel coding is fixed. The figure shows that as the channel error rate increases, a bigger percentage of the total bandwidth should be allocated for the concealment redundancy so as to achieve the best video quality. The error-concealment problem can be formulated loosely as designing a pair of source coder/decoder and transport coder/decoder so that the signal distortion at the decoder is minimized with a given video source model, total channel bandwidth, and channel error characteristics.

The above problem is very difficult, if not impossible, to solve due to the many involved variables and the fact that it is often difficult to model or describe these variables. First, the design of a source coder requires a good model of the source to improve its performance in terms of both coding efficiency and robustness to transmission errors. However, natural video sources are highly nonstationary in nature, and no effective model has been found. In addition, error characteristics of some video transmission channels are also nonstationary and can change significantly during a service session. For example, an ATM network can become congested with the use of statistical multiplexing for a large number of sources, among other reasons. A mobile video phone may operate at dramatically different error rates depending on weather conditions, vehicle moving speeds, etc. Furthermore, other factors such as processing delay, implementation complexity, and application configuration further make the problem difficult to solve.

There have been many techniques proposed in the literature that attack the transmission error problem from different angles. In most if not all cases, some of the variables are fixed first, and then a locally optimal solution is obtained. In this paper, we categorize these techniques into three groups by whether the encoder or decoder plays the primary role or both are involved in cooperation. *Forward error concealment* refers to those techniques in which the encoder plays the primary role. In these techniques, the source-coding algorithm and/or transport-control mechanisms are designed either to minimize the effect of transmission errors without requiring any error concealment at the decoder or to make the error-concealment task at the decoder more effective. Examples of forward error concealment include FEC, joint source and channel coding, and layered coding. On the other hand, *error concealment by postprocessing* includes techniques in which the decoder fulfills the task of error concealment. In general, these methods attempt to recover the lost information by estimation and interpolation without relying on additional information from the encoder. Spatial and temporal smoothing, interpolation, and filtering fall into this category. Last, if the encoder and decoder work cooperatively to minimize the impact of transmission errors, the underlying techniques are called *interactive error concealment*. Examples in this category include ARQ and selective predictive coding based on feedback from the decoder.

Before delving into the details of various techniques, it is worthwhile to mention the criteria for judging their pros and cons. Obviously, the effectiveness of a technique in terms of image quality is the most important. The required delay is also critical for two-way and multipoint transmission. The third factor is the bit-rate overhead incurred by the added concealment redundancy at the source and/or transport level. Last, the processing complexity is always an issue for any system. Note that the priority of these criteria may change depending on the underlying application. For example, delay is much less important for one-way video transmission such as Internet video streaming and video on demand than for two-way and

multipoint video conferencing. In addition, some of the techniques can work for one specific application only, while others may be applied to or adapted to suit a broad range of applications. For instance, retransmission may work well for point-to-point transmission, but it is difficult to use in multipoint applications. On the other hand, error concealment by decoder postprocessing can be applied in almost any application.

III. ERROR DETECTION

Before any error-concealment technique can be applied at the decoder, it is necessary first to find out whether and where a transmission error has occurred. In this section, we review some of the techniques developed for this purpose. We divide these techniques into two categories: those performed at the transport coder/decoder and those at the video decoder.

One way to perform error detection at the transport coder is by adding header information. For example, in packet-based video transmission, the output of the video encoder is packetized into packets, each of which contains a header and payload field [10]. The header further contains a sequence number subfield that is consecutive for sequentially transmitted packets. At the transport decoder, the sequence number can be used for packet-loss detection. For example, the multiplex standard H.223 uses such a method for packet-loss detection [6].

Another method for error detection at the transport level is to use FEC [11]. In this method, error-correction encoding is applied to segments of the output bit stream of the encoder. At the decoder, error-correction decoding is employed to detect and possibly correct some bit errors. For example, H.223 uses FEC for both the multiplex packet header and payload to detect errors in the header and payload, respectively [6]. In H.261, an 18-bit FEC code is applied to each video transport frame of 493 bits for error detection and correction [1].

To accomplish error detection at the video decoder, characteristics of natural video signals have also been exploited. In the methods proposed in [12] and [13], differences of pixel values between two neighboring lines are used for detecting transmission errors in pulse code modulation (PCM) and differential (D)PCM coding. When the difference is greater than a threshold, the current image segment is declared to be damaged. In [14], Mitchell and Tabatabai proposed to detect the damage to a single DCT coefficient by examining the difference between the boundary pixels in a block and its four neighbor blocks. At the decoder, four separate difference vectors are formed by taking the differences between the current block and its adjacent blocks over the 1-pixel-thick boundary in four directions, respectively. Then a one-dimensional (1-D) DCT is applied to these difference vectors. Assuming that the transition between blocks is smooth, the values of the 1-D DCT vectors should be relatively small in the absence of transmission errors. Hence, if these vectors have a

dominant coefficient, then it is declared that one coefficient² is damaged after some statistic test. In addition, the position of the damaged coefficient is also estimated.

Lam and Reibman studied the problem of error detection in the frequency domain [15]. With this approach, a synchronization code word is inserted at the end of each scan line of blocks. When a synchronization code word is captured at the end of a scan line, the number of blocks decoded is checked against a predetermined number. If a difference is found, then an error is declared and the position of the erroneous block is determined as follows. A weighted mean squared error is calculated between the coefficients of each block in the current line and that in the previous line for an 8×8 block. A larger weight is used for low-frequency coefficients and a smaller weight for high-frequency coefficients, so that the distortion measure correlates more closely to the human visual system. The block with the maximum error is recognized as the erroneous block. This block is split into two blocks or merged with an adjacent block, depending on whether the number of blocks decoded is smaller or larger than the prescribed number. When multiple blocks are damaged, the above detection and splitting/merge procedure repeats until the number of blocks matches the desired one.

As mentioned previously, when VLC is used in the source coder, any damage to a single bit can cause desynchronization, resulting in the subsequent bits' being undecodable. However, this property can be used as a means to detect transmission errors. Note that in most cases, the VLC being used is not a complete code, i.e., not all the possible code words are legitimate code words. Hence, once a video decoder detects a code word that is not in its decoding table, a transmission error is declared. In addition, the syntax embedded in the bit stream can also be used for error detection. For example, if the decoded quantization step size is zero, or the number of decoded DCT coefficients is more than the maximum number of coefficients (for example, 64 for an 8×8 DCT transform coder), then a transmission error is detected.

Generally, error detection by adding header information and/or FEC codes at the transport level is more reliable, albeit at the expense of additional channel bandwidth. The benefit of error-detection techniques at the video decoder that rely on the smoothness property of video signals is that they do not add any bits beyond that allocated to the source coder. The use of synchronization code words and/or incomplete VLC codes offers a compromise: by retaining a small degree of redundancy in the encoding process, it eases the error detection at the decoder. Obviously, these techniques are not mutually exclusive and can be employed jointly in practical systems.

IV. FORWARD ERROR CONCEALMENT

In the previous section, we reviewed techniques for detecting transmission errors. From this section onwards,

²This scheme assumes that at most, one coefficient is damaged. In the event that multiple coefficients are damaged, the algorithm detects and corrects only the coefficient that has the largest error.

we will assume that the locations of errors are known and discuss techniques for concealing the detected errors. In this section, we describe error-concealment techniques in which the encoder plays the primary role. When the transport channel is not lossless, there are two kinds of distortion observed at the decoder. The first is the quantization noise introduced by the waveform coder. The second is the distortion due to transmission errors. An optimal pair of source coder and transport coder (including FEC, packetization, and transport protocols) should be designed such that the combined distortion due to both quantization and transmission errors is minimized, given the available bandwidth and channel error characteristics. Typically, the video codec is designed to minimize the quantization error given the available bandwidth. This practice is guided by the well-known source-channel separation theorem of Shannon, which states that one can separately design the source and channel coder to achieve the optimal performance of the overall system. This result was first shown by Shannon for source and channels that are memoryless and stationary [16] and was later extended to a more general class of sources and channels [17]. However, this theorem assumes that the complexity and processing delay of the source and channel coder can be infinite. In most real-world applications, the above assumptions are not true. First, both the source signals and channel environments can vary rapidly and hence are nonstationary. Second, source and channel coders have to be implementable with acceptable complexity and delay. In this situation, joint design of source and channel coder (more generally, transport coder) may achieve better performance.

There are many ways to accomplish forward error concealment. Essentially, they all add a controlled amount of redundancy in either the source coder or the transport coder. In the first case, the redundancy can be added in either the waveform coder or the entropy coder. Some techniques require cooperation between the source and transport coders, while others merely leave some redundancy in or add auxiliary information to the coded data that will help error concealment at the decoder. Some techniques require the network to implement different levels of quality of service (QoS) control for different substreams, while others assume parallel equal paths. In the following, we review these approaches separately.

A. Layered Coding with Transport Prioritization

Until now, the most popular and effective scheme for providing error resilience in a video transport system has been layered coding combined with transport prioritization.³ In layered coding, video information is partitioned into more than one group or layer [3], [18]–[24]. Fig. 4 shows the block diagram of a generic two-layer coding and transport system. The base layer contains the essential information for the video source and can be used to generate

³The term transport prioritization here refers to various mechanisms to provide different QoS in transport, including using unequal error protection, which provides different channel error/loss rate, and assigning different priorities to support different delay/loss requirements.

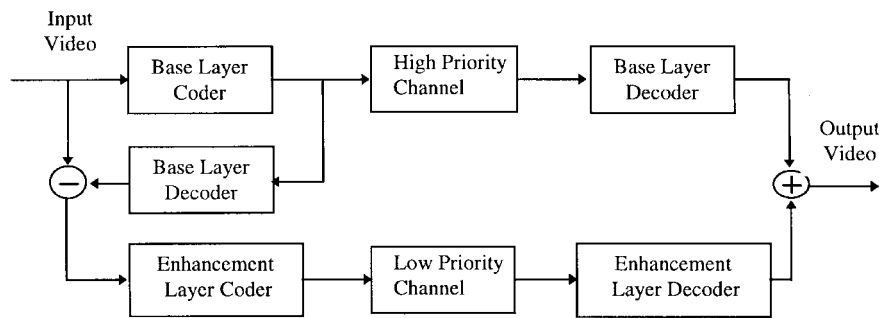


Fig. 4. Block diagram of a system using layered coding and prioritized transport.

an output video signal with an acceptable quality. With the enhancement layers, a higher quality video signal can be obtained. To combat channel errors, layered coding must be combined with transport prioritization so that the base layer is delivered with a higher degree of error protection. Different networks may implement transport prioritization using different means. In ATM networks, there is one bit in the ATM cell header that signals its priority. When traffic congestion occurs, a network node can choose to discard the cells having low priority first. Transport prioritization can also be implemented by using different levels of power to transmit the substreams in a wireless transmission environment. This combination of layered coding with unequal power control has been studied for video transmission in wireless networks [22], [23]. In addition, prioritization can be realized with using different error-control treatments to various layers. For example, retransmission and/or FEC can be applied for the base layer, while no or weaker retransmission/FEC may be applied to the enhancement layers. This approach was taken in the wireless video transport system proposed in [23].

Layered coding can be implemented in several different fashions depending on the way the video information is partitioned. When the partition is performed in the temporal domain, the base layer contains a bit stream with a lower frame rate, and the enhancement layers contain incremental information to obtain an output with higher frame rates. In spatial domain layered coding, the base layer codes the subsampled version of the original video sequence and the enhancement layers contain additional information for obtaining higher spatial resolution at the decoder. The base layer can also encode the input signal with a coarser quantizer, leaving the fine details to be specified in the enhancement layers. In general, it can be applied to the input samples directly or the transformed samples. We refer the first two techniques as *temporal* and *spatial resolution refinement*, respectively, and the third one as *amplitude resolution refinement*. Last, in transform or subband based coders, one can include the low-frequency coefficients or low-frequency band subsignals in the base layer while leaving the high-frequency signal in the enhancement layer. We call this technique *frequency-domain partitioning*. In a video coder using motion-compensated prediction, the coding mode and motion vectors are usually put into the base layer since they are the most important information. Note

that the above schemes do not have to be deployed in isolation; rather, they can be used in different combinations. The MPEG-2 video-coding standard provides specific syntax for achieving each of the above generic methods. In MPEG-2 terminology, layered coding is referred to as *scalability*, and the above four types of techniques are known as *temporal scalability*, *spatial scalability*, *signal-to-noise-ratio (SNR) scalability*, and *data partitioning*, respectively [3].

Although no explicit overhead information is added in layered coding, the graceful degradation of the image quality in the presence of transmission errors is obtained by trading off the compression gain and system complexity. In general, both the encoder and the decoder have to be implemented with the more complicated multilayer structure. In addition, layering will add more coding overhead in the source coder and the transport layer. The coding overhead depends on several factors, including the layered coding method, source spatial and temporal resolution, and bit rate. For example, with the data partition method, a relatively lower overhead will be needed at a higher bit rate than that at a lower bit rate. The four methods presented above have different tradeoffs between robustness to channel noise and coding gain. The study in [24] has found that the three scalability modes in MPEG-2—namely, data partitioning, SNR scalability, and spatial scalability—have increasingly better error robustness, in that order, but also increasing coding overhead. To be more precise, data partitioning requires the least number of bits (requiring only 1% more bits than a single-layer coder at the bit rate of 6 Mbps) to achieve the same image quality when both layers are error free, while the spatial scalability has a better reconstructed image when there exist significant losses in the enhancement layer. SNR scalability is in the middle on both scales. Compared to the one-layer coder, the coder performance is improved significantly over the one-layer coder in presence of channel errors at a relatively small amount of overhead. Table 1 summarizes the required ratio of the base layer to the total bit rate and the highest packet loss rate at which the video quality is still considered visually acceptable. These results are obtained by assuming that the base layer is always intact during the transmission.

When designing a layered coder, a factor that needs to be taken into account is whether the information from the enhancement layers will be used for the prediction in the base-layer coding. When it is used, the coding

Table 1 Comparison of Different Scalability Modes in the MPEG-2 Video Coder (Summarized from Experimental Results Reported in [24])

| Coding Mode | Required base layer to total bit rate ratio | The maximum sustainable packet loss rate |
|---------------------|---|--|
| One Layer (MP@ML) | N/A | 10^{-5} |
| Data Partitioning | 50% | 10^{-4} |
| SNR Scalability | <20% | 10^{-3} |
| Spatial Scalability | <20% | $10^{-3} \sim 10^{-2}$ |

gain in the base layer will be improved. But when the enhancement information is lost during transmission, it will cause distortion in the base layer in addition to the distortion in the enhancement layer. Hence, in some systems, the base-layer prediction is performed with information from the base layer only in order to prevent this prediction memory mismatch in the base layer [18].

B. Multiple-Description Coding

As described in Section IV-A, layered coding can offer error resilience when the base layer is transmitted in an essentially error-free channel, realized via strong FEC and retransmission. In certain applications, however, it may not be feasible or cost effective to guarantee lossless transmission of a certain portion of the transmitted data. In this case, a loss in the base layer can lead to a disastrous effect in the decoded visual quality. An alternative approach to combat transmission errors from the source side is by using *multiple-description coding* (MDC). This coding scheme assumes that there are several parallel channels between the source and destination and that each channel may be temporarily down or suffering from long burst errors. Furthermore, the error events of different channels are independent, so that the probability that all channels simultaneously experience losses is small. These channels could be physically distinct paths between the source and destination in, for example, a wireless multihop network or a packet-switched network. Even when only one single physical path exists between the source and destination, the path can be divided into several virtual channels by using time interleaving, frequency division, etc.

With MDC, several coded bit streams (referred to as “descriptions”) of the same source signal are generated and transmitted over separate channels. At the destination, depending on which descriptions are received correctly, different reconstruction schemes (or decoders) will be invoked. The MDC coder and decoder are designed such that the quality of the reconstructed signal is acceptable with any one description and that incremental improvement is achievable with more descriptions. A conceptual schematic for a two-description coder is shown in Fig. 5. In this case, there are three decoders at the destination, and only one operates at a time. To guarantee an acceptable quality with a single description, each description must carry sufficient information about the original signal. This implies that there will be overlap in the information contained in different descriptions. Obviously, this will reduce the coding ef-

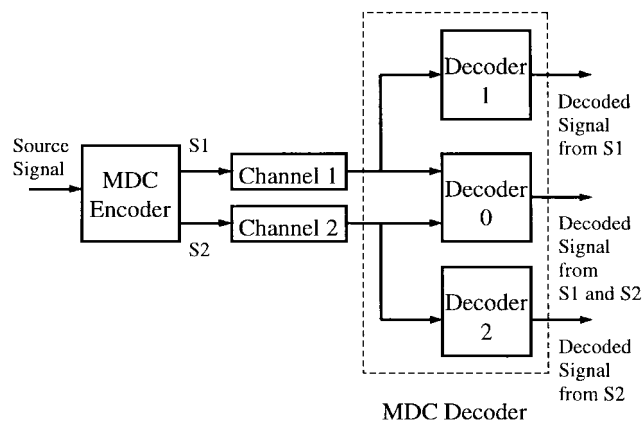


Fig. 5. Illustration of multiple description coding and decoding.

iciency compared to the conventional single description coder (SDC) that is aimed at minimizing the distortion in the absence of channel loss. This has been shown using a rate-distortion analysis for different types of sources [25]–[27]. However, this reduced coding efficiency is in exchange for increased robustness to long burst errors and/or channel failures. With SDC, one would have to spend many error-control bits and/or introduce additional latency (in all the bits or only the base layer in the layered coding case) to correct such channel errors. With MDC, a long burst error or even the loss of an entire description does not have a catastrophic effect as long as not all the substreams experience failure simultaneously. Thus, one could use fewer error-control bits for each substream.

A simple way of obtaining multiple equally important descriptions is by splitting adjacent samples among several channels using an interleaving subsampling lattice and then coding the resulting subimages independently [28]–[31]. If one subimage is lost, it can be recovered satisfactorily based on correlation among adjacent samples in the original image. This approach requires a quite large bit-rate overhead because the coder cannot make use of the correlation among adjacent samples. In the following, we review two other approaches that are more efficient.

1) *Multiple-Description Scalar Quantization (MDSQ)*: In the approach of Vaishampayan [30], two substreams are obtained by producing two indexes for each quantized level. The index assignment is designed so that if both indexes are received, the reconstruction accuracy is equivalent to a fine quantizer. On the other hand, if only one index is received, the reconstruction accuracy is essentially that of a

coarse quantizer. A simple implementation of this approach is by using two quantizers whose decision regions shift by half of the quantizer interval with respect to each other (known as A2 index assignment [30]). If each quantizer has a bit rate of R , the reconstruction error from two descriptions (i.e., both indexes for each quantized sample) is equivalent to that of a single $R + 1$ bit quantizer. On the other hand, if only one description is available, the performance is equivalent to that of a single R bit quantizer. In the absence of channel failure, a total of $2R$ bits are required to match the performance of a single quantizer with $R + 1$ bits. Therefore, the loss of coding efficiency is quite significant for large values of R . At lower bit rates, the overhead is smaller. More sophisticated quantizer mappings can be designed to improve the coding efficiency. The MDSQ approach is first analyzed assuming both index streams are coded using fixed-length coding [30]. It is later extended to consider entropy coding of the indexes [31]. The original MDSQ approach is developed for memoryless sources. To handle sources with memory, MDSQ can be embedded in a transform coder by coding each transform coefficient using MDSQ [32], [33]. This approach has been applied to transform-based image and video coders.

2) *MDC Using Correlation-Inducing Linear Transforms:* Another way of introducing correlation between multiple streams is by linear transforms that do not completely decorrelate the resulting coefficients. Ideally, the transform should be such that the transform coefficients can be divided into multiple groups so that the coefficients between different groups are correlated. This way, if some coefficient groups are lost during transmission, they can be estimated from the received groups. To minimize the loss of coding efficiency, the coefficients within the same group should be uncorrelated. To simplify the design process for a source signal with memory, one can assume the presence of a prewhitening transform so that the correlation-inducing transform can operate on uncorrelated samples.

In [34] and [35], Wang *et al.* and Orchard *et al.*, respectively, proposed applying a pair-wise correlating transform (PCT) to each pair of uncorrelated variables obtained from the Karhunen-Loeve transform (KLT). The two coefficients resulting from the PCT are split into two streams that are then coded independently. If both streams are received, then an inverse PCT is applied to each pair of transformed coefficients, and the original variables can be recovered exactly (in the absence of quantization errors). If only one stream is received, the missing stream can be estimated based on the correlation between the two streams. In [34], the PCT uses a 45° rotation matrix, which yields two coefficients having equal variance and therefore requiring the same number of bits. More general classes of PCT using any rotation matrix (i.e., orthogonal) as well as nonorthogonal matrices are considered in [35]. The overhead introduced by this approach can be controlled by the number of coefficients that are paired, the pairing scheme, and the transform parameters (e.g., the rotation angle). This method

has been integrated in a JPEG⁴-like coder, in which the PCT is applied to the DCT (similar to KLT in decorrelation capability) coefficients. Only the 45° rotation case has been simulated. It is shown that to guarantee a satisfactory quality from one stream, about 20% overhead is required over the JPEG coder for 512×512 images coded at about 0.6 bits per pixel (bpp). As noted before for layered coding, the overhead rate depends on the type of image being coded and the reference bit rate.

Instead of designing the transform basis functions to introduce correlation among coefficients in the same block, an alternative approach is to introduce correlation among the same coefficients in different blocks. Then one can obtain multiple descriptions by splitting coefficients in adjacent blocks into separate descriptions and recover lost coefficient blocks in one description from the coefficient blocks in the other description by making use of the interblock correlation. To introduce additional correlation beyond what is originally present between adjacent image blocks, overlapping block transforms can be used. In [36], Hemami designed a family of lapped orthogonal transform (LOT) bases, referred to as T6–T9, which achieve different tradeoffs between compression gain and reconstruction gain. The latter is defined for a special case in which a lost block is recovered by the mean of four adjacent blocks. Recently, Chung and Wang [37] developed a reconstruction method that can achieve significantly better reconstruction quality than the mean reconstruction method. The method is based on the principle of maximally smooth recovery, to be introduced in Section V-B. It was found that with the T6 basis, satisfactory image quality can be guaranteed from a single description alone (including every other block) at a bit rate overhead of 0.3–0.4 bpp over that achievable by the LOT-DCT basis, which is designed to maximize the coding efficiency [38]. Interestingly, their study found that the required number of additional bits is fairly constant among different images, so that the relative overhead is lower for an image requiring a high bit rate. For the image “Lena,” the relative overhead is about 30–40%, while for the more complex image “People,” which is a snapshot of a crowded audience, it is merely 10%.

Given the relatively large overhead associated with MDC, this approach is appropriate only for channels that have relatively high loss or failure rates. When the channel loss rate is small, the reconstruction performance in the error-free case dominates, and the SDC, which is optimized for this scenario, performs best. On the other hand, when the loss rate is very high, the reconstruction quality in the presence of loss is more critical, so that the MDC approach becomes more suitable. For example, it has been found that under the same total bit rate, the reconstruction quality obtained with the transform coder using PCT exceeds that of the JPEG coder (with even and odd blocks split among two channels) only when the block loss rate is larger than 10^{-3} [39]. Similarly, in the MDC coder using LOT, the T6–T9 bases were shown to provide better reconstruction

⁴JPEG is a standard of the Joint Photographic Experts Group of the ISO.

quality than the DCT-LOT basis only when the block loss probability is larger than 0.025 [37]. A challenging task is how to design the MDC coder that can automatically adapt the amount of added redundancy according to underlying channel error characteristics.

C. Joint Source and Channel Coding

In layered coding and MDC, introduced previously, the interaction between the source and channel coders is exercised at a high level. In layered coding, the source coder produces a layered stream assuming that the channel coder can guarantee the delivery of the most important source layer. On the other hand, with MDC, the source coder assumes that all coded bits will be treated equally and that all are subject to loss. In this section, we review techniques that invoke the source-channel interaction at a lower level, i.e., the quantizer and entropy-coder design at the source coder and the design of FEC and modulation schemes at the channel coder. This type of approach is traditionally referred to as *joint source and channel coding*, although in a broader sense, layered coding and MDC can also be considered to belong to this category.

In general, joint source and channel coding is accomplished by designing the quantizer and entropy coder for given channel error characteristics to minimize the effect of transmission errors. Spilker noted that when the channel becomes very noisy, a coarse quantizer in the source-coding stage outperforms a fine quantizer for a PCM-based source coder [40]. Kurtenbach and Wintz designed optimal quantizers to minimize the combined mean square error introduced by both quantization and channel errors given the input data probability distribution and the channel error matrix [41]. Farvardin and Vaishampayan further extended the design of the optimal quantizer and also proposed a method for performing the code-word assignment to match the channel error characteristics [42].

The above studies were conducted for a general source. For image signals, Modestino and Daut first considered the application of convolution codes to protect against channel errors following a source coder using DPCM [43]. This technique was later extended to transform coders using DCT [44]. Three options were proposed to implement combined source and channel coding. In the first option, modulation and ECC are the same for all the bits in every quantized transform coefficient. In the second option, modulation and ECC are the same for all the bits belonging to the same quantized coefficient but can be different for different coefficients. In the third option, modulation and FEC are allowed to vary among different bits of the same coefficient. It was shown that with the first option, for a typical outdoor image, when the channel SNR is smaller than 10 dB, the SNR for the received picture is better with 50% error-correction bits than without any error correction bits. The second and third options can further extend the channel SNR threshold to below 5 dB [44]. Vaishampayan and Farvardin considered the adaptation of bit allocation (assuming fixed-length coding) for DCT coefficients based on channel error characteristics [45]. The basic conclusion

was that for noisier channels, fewer bits should be allocated to the high-frequency coefficients and more bits should be allocated to the low-frequency coefficients.

D. Robust Waveform Coding

In traditional source-coder design, the goal is to eliminate both the statistical and the visual redundancy of the source signal as much as possible to achieve the best compression gain. This, however, makes the error-concealment task at the decoder very difficult. One approach to solve this problem is by intentionally keeping some redundancy in the source-coding stage such that better error concealment can be performed at the decoder when transmission errors occur. We refer to techniques in this group as *robust waveform coding*. Strictly speaking, layered coding and MDC both belong to this category, as they both add some redundancy in the coded bit streams to provide the robustness to channel errors. The techniques described in this subsection assume that the coded source signal will be transmitted in a single channel. They can be applied to produce individual streams in layered coding and MDC.

1) *Adding Auxiliary Information in the Waveform Coder:* One simple approach to combat transmission errors is by adding auxiliary information in the waveform coder that can help signal reconstruction in the decoder. As will be shown in Section V-A, an effective technique for error concealment in the decoder is the use of motion-compensated temporal interpolation. This requires knowledge of the motion vectors of the lost blocks. One way to help the error-concealment task is by sending motion vectors for macroblocks that would not ordinarily use motion-compensated prediction. For example, in MPEG-2, the coder has the option of sending motion vectors for macroblocks in I-frames, so that I-frames can be recovered reliably [3]. In the absence of channel errors, these motion vectors are useless. However, when certain macroblocks in an I-frame are damaged, their motion vectors can be estimated from those of the surrounding received macroblocks, and then these macroblocks can be recovered from the corresponding motion-compensated macroblocks in the previous frame.

In [46], Hemami and Gray proposed to add some auxiliary information in the compressed bit stream so that the decoder can interpolate lost image blocks more accurately. A damaged image block is interpolated at the decoder using a weighted sum of its correctly received neighbor blocks. Determination of the interpolation coefficients is combined with vector quantization in a single step at the encoder, and the resulting quantized weights are transmitted as overhead information, which is less than 10% for typical JPEG-coded images.

2) *Restricting Prediction Domain:* To reduce the effect of error propagation due to the use of prediction, one can limit the prediction within nonoverlapping spatial and temporal regions. For example, the H.263 standard divides a picture into *slices*, and in the *independent segment decoding* mode, spatial and temporal prediction is confined within each slice. Here, spatial prediction refers to prediction of DCT coefficients and motion vectors of one macroblock

from adjacent macroblocks, and temporal prediction is the well-known motion-compensated interframe prediction. To further suppress the effect of temporal error propagation, in the H.263 standard, input video frames are partitioned into separate groups called *threads*, and each thread is coded without using other threads for prediction [47]. This is referred to as *video redundancy coding*. For example, when two threads are used, the even and odd frames are grouped separately, and temporal prediction is performed within each group. All the threads start from the same sync-frame (for example, I-frame) and end at another sync-frame. When a transmission error occurs in one frame, only one thread will be affected. Between the affected frame and the next sync-frame, a video signal with half the frame rate can be produced. Obviously, restricting the prediction domain as described above will reduce the coding efficiency. But it will also confine the picture-quality degradation to only one region when a transmission error occurs. Therefore, this is another way to trade off coding gain for better reconstructed picture quality.

E. Robust Entropy Coding

In the techniques described in Section IV-D, redundancies are added during the waveform coding stage. One can also add redundancy in the entropy-coding stage to help detect bit errors and/or prevent error propagation. We call such techniques *robust entropy coding*. In this section, we first review techniques that use synchronization code words to limit error propagation in compressed data and then describe several VLC codes that are designed to be error resilient.

1) *Self-Synchronizing Entropy Coding*: When VLC is used in video coding, a single bit error can lead to the loss of synchronization. First, the decoder may not know that a bit error has happened. Furthermore, even when the decoder recognizes that an error has occurred by other means such as the underlying transport protocol, it may not know which bit is in error and hence it cannot decode the subsequent bits. One way to prevent this is to designate one code word as the synchronization code word in the entropy coder [48]–[52]. A synchronization code word has the property that the entropy decoder will regain synchronization once a decoder captures such a code word. Generally, the resulting entropy coder will be less efficient in terms of the compression ratio than the “optimal” coder without using the synchronization code word.

Although synchronization can be obtained with a synchronization code word, the number of decoded symbols may be incorrect. This will typically result in a shift of sequential blocks in a block-based coder. To solve this problem, a distinct synchronization code word can be inserted at a fixed interval either in the pixel domain [51], [52] or in the bit-stream domain where the number of coded bits is used for measuring the interval [53], [54]. Side information such as spatial and temporal locations is normally included after the synchronization code word to identify where the decoded blocks belong. The synchronization code word in this case does not carry any information

on the encoded video but only plays the role of enabling the decoder to regain synchronization. Several methods have been proposed to minimize the bit-rate overhead introduced by the synchronization code word [51], [52]. While a shorter synchronization code word introduces less overhead, it also increases the probability that a bit error may generate a fake synchronization code word. Hence, in practical video-coding systems such as H.261 and H.263, relatively long synchronization code words are used instead [1], [2].

For high-error-rate environments such as wireless networks, MPEG-4 allows the insertion of an additional synchronization code word, known as *motion marker*, within each coded block between the motion information and the texture information [53], [54]. When only the texture information is damaged, then the motion information for a block can be still used for better error concealment with techniques to be described in Section V.

2) *Error-Resilient Entropy Coding (EREC)*: With the methods described above, error propagation is limited to the maximum separation between the synchronization code words. To reduce the introduced redundancy, however, these codes have to be used infrequently. Kingsbury *et al.* have developed EREC methods [55], [56]. In the method of [56], variable-length bit streams from individual blocks are distributed into slots of equal size. Initially, the coded data for each image block are placed into the designated slot for the block either fully or partially. Then, a predefined offset sequence is used to search for empty slots to place any remaining bits of blocks that are bigger than the slot size. This is done until all the bits are packed into one of the slots. With EREC, the decoder can regain synchronization at the start of each block. It also ensures that the beginning of each block is more immune to error propagation than those at the end. This way, error propagation is predominant only in higher frequency coefficients. The redundancy introduced by using EREC is negligible. In [56], when EREC is integrated into an H.261 like coder, the reconstruction quality at the bit error rate (BER) of 10^{-4} – 10^{-3} is significantly better. Recently, the above EREC method has been used to transcode an MPEG-2 bit stream to make it more error resilient [57]. With additional enhancement and error concealment, the video quality at a BER of 10^{-2} was considered acceptable. Kawahara and Adachi also applied the EREC method at the macroblock level together with unequal error protection for H.263 transmission over wireless networks [58]. Their simulation results show that the proposed method outperforms the plain FEC both for random bit errors at BER greater than 10^{-3} and for burst errors.

In the error-resilient mode of MPEG-4 [53], [54], reversible variable-length code (RVLC) is employed, which can make full use of the available data when a transmission error occurs. RVLC is designed in such a way that once a synchronization code word is found, the coded bit stream can be decoded backward. With conventional VLC, all data after an erroneous bit are lost until the next synchronization code word. On the other hand, RVLC can recover data

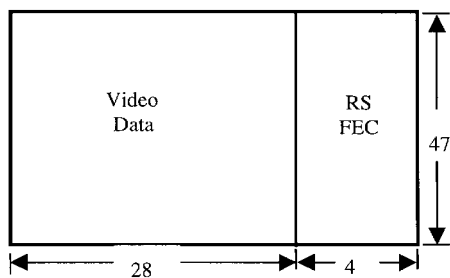


Fig. 6. Illustration of FEC with interleaving for ATM cell-loss recovery. The numbers in the figure are in bytes.

backward from the next synchronization code word detected until the first decodable code word after the erroneous bit. This improved robustness is again achieved at a reduced coding efficiency, which is due to the constraint imposed by constructing the RVLC tables.

F. FEC Coding

FEC is well known for both error detection and error correction in data communications. FEC has the effect of increasing transmission overhead and hence reducing usable bandwidth for the payload data. Therefore, it must be used judiciously in video services that are very demanding in bandwidth but can tolerate a certain degree of loss. FEC has been studied for error recovery in video communications [59]–[64]. In H.261, an 18-bit error-correction code is computed and appended to 493 video bits for detection and correction of random bit errors in integrated services digital network (ISDN). For packet video, it is much more difficult to apply error correction because several hundred bits have to be recovered when a packet loss occurs.

In [59], Lee *et al.* proposed to combine Reed–Solomon (RS) codes with block interleaving to recover lost ATM cells. As shown in Fig. 6, an RS (32,28,5) code is applied to every block of 28 bytes of data to form a block of 32 bytes. After applying the RS code row by row in the memory up to the forty-seventh row, the payload of 32 ATM cells is formed by reading column by column from the memory with the attachment of one byte indicating the sequence number. In this way, a detected cell loss at the decoder corresponds to one byte erasure in each row of 32 bytes after deinterleaving. Up to four lost cells out of 32 cells can be recovered. The Grand-Alliance high-definition television broadcast system has adopted a similar technique for combating transmission errors [60]. In addition to using the RS code, data randomization and interleaving are also employed for additional protection.

Because a fixed amount of video data has to be accumulated to perform the block interleaving described above, relatively long delay will be introduced. To reduce the interleaving delay, a diagonal interleaving method is proposed in [63]. At the encoder side, input data are stored horizontally in a designated memory section, which are then read out diagonally to form ATM cells. In the decoder, the data are stored diagonally in the memory and are read out

horizontally. In this way, the delay due to interleaving is halved.

The use of FEC for MPEG-2 in a wireless ATM local-area network has been studied by Ayanoglu *et al.* in [64]. FEC is used at the byte level for random bit error correction and at the ATM cell level for cell-loss recovery. These FEC techniques are applied to both single-layer and two-layer MPEG data. It was shown that the two-layer coder outperforms the one-layer approach significantly, at a fairly small overhead. The paper also compared direct cell-level coding with the cell-level interleaving followed by FEC. It is interesting to note that the paper concludes that the latter introduces longer delay and bigger overhead for equivalent error-recovery performance and suggests that direct cell-level correction is preferred.

G. Transport-Level Control

The forward error concealment techniques reviewed above are exercised at the source coder. Forward error concealment can also be accomplished at the transport level. A good example of this is error isolation by structured packetization schemes in packet video. The output of the source coder is assembled into transport packets in such a way that when a packet is lost, the other packets can still be useful because the header and coding mode information is embedded into successive packets [65], [66].

A packet often contains data from several blocks. To prevent the loss of contiguous blocks because of a single packet loss, interleaved packetization can be used, by which successive blocks are put into nonadjacent packets [19], [61]. This way, a packet loss will affect blocks in an interleaved order (i.e., a damaged block is surrounded by undamaged blocks), which will ease the error-concealment task at the decoder. Note that the use of interleaved packetization in the transport layer requires the source coder to perform block-level prediction only within blocks that are to be packetized sequentially. This will reduce the prediction gain slightly.

Last, the binary bits in a compressed video bit stream are not equally important. When layered coding is used at the source coder, the transport controller must assign appropriate priority to different layers, which is a form of transport-level control. Even with a nonlayered coder, the picture header and other side information are much more important than the block data. These important bits should be protected so that they can be delivered with a much lower error rate. One way to realize this is by using dual transmission of important information. In [67], dual transmission for picture header information and quantization matrix was proposed for MPEG video. In [68], Civanlar and Cash considered video-on-demand services over an ATM network where the servers and the clients are Internet protocol based and are connected to the network via a fiber distributed data interface network. They proposed to use transmission control protocol (TCP) for transmission of a very small amount of high-priority data before a service session and user datagram protocol for the remaining low-priority data during the session.

Table 2 Summary of Forward Error-Concealment Techniques

| |
|--|
| Layered coding with prioritized transport: |
| <ul style="list-style-type: none"> • Frequency domain partitioning (e.g., MPEG2 data partitioning) • Successive amplitude refinement (e.g., MPEG2 SNR scalability) • Spatial/temporal resolution refinement (e.g. MPEG2 spatial/temporal scalability) |
| Multiple description coding: |
| <ul style="list-style-type: none"> • Multiple description scalar quantization [30] • Correlation inducing transforms [34,35] • Spatial domain subsampling [28,31] • Transform domain subsampling [37] |
| Robust waveform coding: |
| <ul style="list-style-type: none"> • Adding auxiliary information to help error concealment [46] • Restricting prediction domain (e.g. independent segment decoding and video redundancy coding in H.263 [2]) |
| Robust entropy coding: |
| <ul style="list-style-type: none"> • Using synchronization codeword to prevent error propagation [49-54] • Error resilient entropy coding [55-58] • Reversible VLC in MPEG4 [54] |
| Joint source and channel coding: |
| <ul style="list-style-type: none"> • Adapt bit allocation and codeword mapping based on channel error characteristics [43-45] |
| Transport level control: |
| <ul style="list-style-type: none"> • Prioritized transport for layered coding • Robust packetization [65,66] • Spatial block interleaving [19] • Dual transmission of important information |

H. Summary

Table 2 summarizes the various techniques that have been developed for forward error concealment. All of these techniques achieve error resilience by adding a certain amount of redundancy in the coded bit streams, at either the source coder or transport coder. Among the techniques that add redundancy at the source coder, some are aimed at guaranteeing a basic level of quality and providing a graceful degradation upon the occurrence of transmission errors (e.g., layered coding and multiple-description coding), some help the decoder to perform error concealment upon detection of errors (e.g., robust waveform coding), while others help to detect bit errors and/or prevent error propagation (e.g., robust entropy coding). The transport-level protection, e.g., by using FEC and robust packetization, etc., must cooperate with the source coder so that more important information bits are given stronger protection and that a single bit error or cell loss does not lead to a disastrous effect. It is noteworthy that some techniques require close interaction between the source and transport coders (e.g., layered coding with prioritized transport, interleaved packetization with restricted prediction domain), while others assume that different substreams are treated equally (e.g.,

multiple-description coding). Note that these techniques are not mutually exclusive; rather, they can be used together in a complementary way.

V. ERROR CONCEALMENT BY POSTPROCESSING AT THE DECODER

It is well known that images of natural scenes have predominantly low-frequency components, i.e., the color values of spatial and temporally adjacent pixels vary smoothly, except in regions with sharp edges. In addition, the human eyes can tolerate more distortion to the high-frequency components than to the low-frequency components. These facts can be used to conceal the artifacts caused by transmission errors. In this section, we describe several techniques that attempt to perform error concealment at the decoder. Some of these techniques can be used in conjunction with the auxiliary information provided by the source coder to improve the reconstruction quality.

Because of the space limit, we will only review methods that have been developed for video coders using block-based motion compensation and nonoverlapping block transforms (the DCT in particular), which is the underlying core technology in all standard video codecs. With such a

coder, a frame is divided into macroblocks, which consists of several blocks. There are typically two coding modes at the macroblock level. In the intramode, each block is transformed using block DCT, and the DCT coefficients are quantized and entropy coded. In the intermode, a motion vector is found that specifies its corresponding macroblock in a previously coded frame,⁵ and this motion vector and the DCT coefficients of the prediction error block are quantized and coded. By using a self-synchronization code word at the beginning of each scan row of macroblocks, known as a slice, typically a transmission error (either a bit error or erasure error) will only cause damage to a single row, so that the upper and lower macroblocks of a damaged block may still be correctly received. If the coded macroblocks are packetized in an interleaved manner, than a damaged macroblock is usually surrounded in all four directions by correctly received macroblocks. In addition, if layered coding with frequency-domain partitioning is used, a damaged macroblock may have the coding mode, motion vector, and some low-frequency coefficients correctly received. Last, the error events among two adjacent frames are usually sufficiently uncorrelated so that for a given damaged macroblock in the current frame, its corresponding macroblock (as specified by the motion vector) in the previous frame is usually received undamaged. All the postprocessing techniques make use of the correlation between a damaged macroblock and its adjacent macroblocks in the same frame and/or the previous frame to accomplish error concealment. Some of the techniques only apply to macroblocks coded in intramode, while others, although applicable to intercoded blocks, neglect the temporal information. In the following, we first review techniques that concentrate on recovery of the DCT coefficients or, equivalently, the pixel values. We then present techniques for recovering the coding mode and motion vectors.

A. Motion-Compensated Temporal Prediction

One simple way to exploit the temporal correlation in video signals is by replacing a damaged macroblock with the spatially corresponding macroblock in the previous frame. This method, however, can produce adverse visual artifacts in the presence of large motion. Significant improvement can be obtained by replacing the damaged macroblock with the motion compensated block (i.e., the block specified by the motion vector of the damaged block). This method is very effective when combined with layered coding that includes all the motion information in the base layer [69]. Because of its simplicity, this method has been widely used. In fact, the MPEG-2 standard allows the encoder to send the motion vectors for intracoded macroblocks, so that these blocks can be recovered better if they are damaged during transmission (refer to Section IV-D1). It has been found that using motion-compensated error

⁵In some coders, bidirectional prediction is used, by which a macroblock in the current frame is predicted from a previous frame and a following frame. To simplify the discussion, we only consider unidirectional prediction here.

concealment can improve the peak SNR of reconstructed frames by 1 dB at a cell-loss rate of 10^{-2} for MPEG-2 coded video [24]. A problem with this approach is that it requires knowledge of the motion information, which may not be available in all circumstances. When the motion vectors are also damaged, they need to be estimated from the motion vectors of surrounding macroblocks, and incorrect estimates of motion vectors can lead to large errors in reconstructed images. Another problem with this approach occurs when the original macroblock was coded with intramode and the coding-mode information is damaged. Then, concealment with this method can lead to catastrophic results in situations such as a scene change. Recovery of motion vectors and coding modes is discussed in Section V-E.

In [70], Kieu and Ngan considered the error-concealment problem in a layered coder that sends the motion vectors and low-frequency coefficients in the base layer and high-frequency coefficients in the enhancement layer. Instead of simply setting the high-frequency components to zero when the enhancement layer is damaged, it was shown that using the high-frequency component from the motion-compensated macroblock in the previous frame can improve the reconstructed picture quality. It is assumed that the base layer is delivered without error. When the enhancement layer is damaged, for each damaged macroblock, its motion-compensated macroblock is formed and the DCT is applied to the blocks within the macroblock. The resulting high-frequency DCT coefficients are then merged with the base-layer DCT coefficients of the damaged blocks in the current frame, and the inverse DCT is applied to these blocks to form an error-concealed macroblock.

The above techniques only make use of temporal correlation in the video signal. For more satisfactory reconstruction, spatial correlation should also be exploited. The techniques reviewed below either make use of both spatial and temporal correlation for error concealment or only exploit the spatial correlation.

B. Maximally Smooth Recovery

This method makes use of the smoothness property of most image and video signals through a constrained energy minimization approach. The minimization is accomplished block by block. Specifically, to estimate the missing DCT coefficients in a block, the method minimizes a measure of spatial and temporal variation between adjacent pixels in this block and its spatially and temporally neighboring blocks, so that the resulting estimated video signal is as smooth as possible. Wang *et al.* first used this approach to recover damaged blocks in still images coded using block-transform-based coders by making use of the spatial smoothness only [71]. Zhu *et al.* later extended this method to video coders using motion compensation and transform coding by adding the temporal smoothness measure [19]. In this latter case, the error function being minimized is a weighted sum of a spatial difference measure and a temporal difference measure. For computational ease, the spatial and temporal difference measures are defined

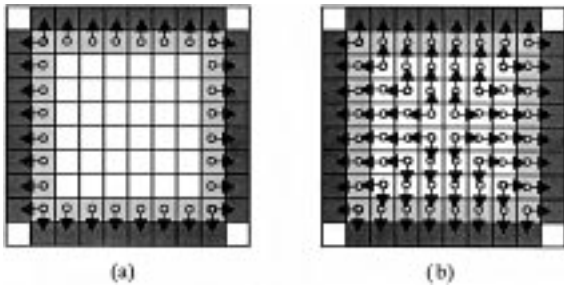


Fig. 7. Illustration of spatial smoothing constraints (from [71]). An arrow between two pixels indicates that the difference between these two pixels is included in the smoothness measure. The measure illustrated in (a) is appropriate when only the DC coefficient is lost, while that in (b) is more effective when the DC and several low-frequency coefficients are missing.

as the sums of squared differences between spatially and temporally adjacent pixels, respectively. Fig. 7 illustrates the two spatial smoothness measures proposed in [71]. To satisfy the constraints imposed by the received coefficients, the image block to be reconstructed is represented in terms of the received coefficients, the missing coefficients to be estimated, and the prediction block in the previous frame (for intercoded blocks only). The solution essentially consists of three linear interpolations—in the spatial, temporal, and frequency domains—from the pixels in adjacent image blocks that have been reconstructed previously, the prediction block in the previous frame, and the received coefficients for this block, respectively. When all the coefficients are lost in a damaged block, the solution reduces to spatial and temporal interpolation only. If one sets the weight for the spatial difference measure to zero, then the solution is equivalent to replacing the damaged block by the prediction block, the same as that presented in Section V-A. On the other hand, if the weighting for the temporal difference measure is zero, only the spatial correlation is used, and the solution is a linear interpolation from the received coefficients and the neighbor pixel data. This can be used for intracoded blocks or still images. The reconstruction operator depends on the weighting factor used and the transform basis functions associated with the lost coefficients. For a given loss pattern (i.e., which coefficients are lost), this operator can be precomputed, and the reconstruction task involves a matrix-and-vector product, with a complexity similar to a block transform.

With the above reconstruction technique, simulation results show that a block with its first 15 low-frequency coefficients lost can be recovered with acceptable quality as long as its neighboring blocks are available for spatial/temporal interpolation⁶ [19]. To improve the robustness of the coder, one can interleave the coefficients of adjacent blocks before transmission so that a channel error will only affect spatially disjointed blocks. Furthermore, the coefficients can be segmented into multiple layers so that only a finite number of loss patterns exist, and the interpolation filters for these loss patterns can be precomputed. These enhancements

⁶Note that if only the high-frequency coefficients are lost, simply setting them to zeros will in general yield satisfactory results.

have been added to an MPEG-1-like video codec, and the reconstruction technique is invoked at the decoder only when the layers containing low-frequency coefficients are lost. Specifically, four layers are used: the base layer contains the coding mode, the second layer includes the motion vectors, and the third and fourth layers carry the low- and high-frequency DCT coefficients, respectively. Simulation results show that this modified MPEG-1 system can yield visually acceptable quality at loss rates of 10^{-3} in the first two layers and 10^{-2} in the third layer [19].

In the above work, the spatial/temporal variation is measured by calculating the difference between two adjacent pixels. Such first-order smoothness measures can lead to blurred edges in the recovered image. Zhu and Wang later investigated the use of second-order smoothness criteria to reduce the blurring artifacts [73]. A combination of the quadratic variation and Laplacian operator was proposed, and the reconstructed images using this measure are visually more pleasing than those obtained with the first-order measure, with sharper edges that are smooth along the edge directions. To further improve the reconstruction quality, an edge-adaptive smoothness measure can be used so that the variation along the edges is minimized but not across the edges. Several techniques have been developed along this direction [72]. This approach requires the detection of edge directions for the damaged blocks. This is a difficult task, and a mistake can yield noticeable artifacts in the reconstructed images. The method using the second-order smoothness measure is in general more robust and can yield satisfactory images with lower computational cost.

C. Projection onto Convex Sets (POCS)

The method described in the previous section makes use of the smoothness property of the image and video signals by an energy minimization approach. An alternative is to use the POCS method. Sun and Kwok proposed to use this method to restore a damaged image block in a block transform coder [74]. The convex sets are derived by requiring the recovered block to have a limited bandwidth either isotropically (for a block in a smooth region) or along a particular direction (for a block containing a straight edge). With this method, a combined block is formed by including eight neighboring blocks with the damaged block. First, this combined block is subject to an edge existence test by using the Sobel operator. The block is classified as either a monotone block (i.e., with no discernible edge orientations) or an edge block. The edge orientation is quantized to one of the eight directions equally spaced in the range of $0-180^\circ$. Then, two projection operators are applied to the combined block, as shown in Fig. 8. The first projection operator implements a band-limitedness constraint, which depends on the edge classifier output. If the block is a monotone block, then the block is subject to an isotropic band-limitedness constraint, accomplished by an isotropic low-pass filter. On the other hand, if the block classifier output is one of the eight edge directions, then a bandpass filter is applied along that direction. The filtering operation is implemented in the Fourier trans-

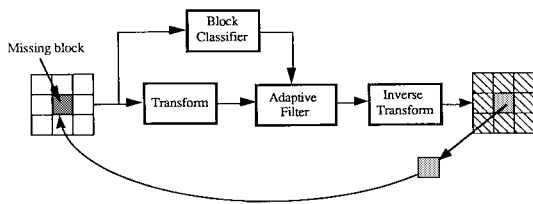


Fig. 8. Illustration of the adaptive POCS iterative restoration process (from [74]).

form domain. The second projection operator implements a range constraint and truncates the output value from the first operator to the range of $[0,255]$. For pixels in the edge blocks that are correctly received, their values are maintained. These two projection operations are applied alternately until the block does not change any more under further projections. It was found that five to ten iterations are usually sufficient when a good initial estimate is available. Note that this technique only makes use of spatial information in the reconstruction process and is therefore applicable to intracoded blocks or still images. For intercoded blocks, one way to make use of the temporal information is by using the motion-compensated block in the previous frame as the initial estimate and then using the technique presented here to further improve the reconstruction accuracy.

D. Spatial- and Frequency-Domain Interpolation

One implication of the smoothness property of the video signal is that a coefficient in a damaged block is likely to be close to the corresponding coefficients (i.e., with the same frequency index) in spatially adjacent blocks. In [75], Hemami and Meng proposed to interpolate each lost coefficient in a damaged block from its corresponding coefficients in its four neighbor blocks. The interpolation coefficients are estimated by minimizing a spatial difference measure given in [71]. When all the coefficients for the damaged block are lost, this frequency-domain interpolation is equivalent to interpolating each pixel in the block from the corresponding pixels in four adjacent blocks rather than the nearest available pixels. Because the pixels used for interpolation are eight pixels away in four separate directions, the correlation between these pixels and the missing pixel is likely to be small, and the interpolation may not be accurate. To improve the estimation accuracy, Aign and Fazel proposed to interpolate pixel values within a damaged macroblock from its four 1-pixel-wide boundaries [76]. Two methods are proposed to interpolate the pixel values. In the first method, a pixel is interpolated from two pixels in its two nearest boundaries, as shown in Fig. 9(a). In the second method, shown Fig. 9(b), a pixel in the macroblock is interpolated from the pixels in all four boundaries.

As with the POCS method, the above schemes only make use of the spatial smoothness property and are mainly targeted for still images or for intracoded blocks in video. For intercoded frames, the frequency-domain interpolation method of [75] cannot be applied because the DCT coeffi-

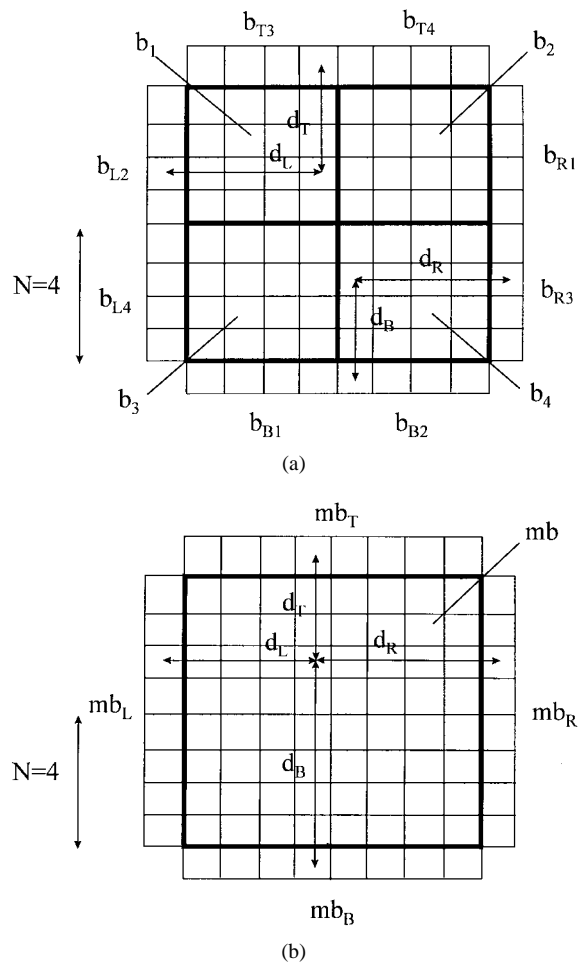


Fig. 9. Spatial interpolation for error concealment. (a) Block based. (b) Macroblock based (adapted from Figs. 1 and 2 in [76]).

icients of prediction errors in adjacent blocks are not highly correlated. The spatial-domain interpolation can, however, be applied to the original pixel values (not the prediction error values).

Due to the smoothness properties of natural images, the correlation between high-frequency components of adjacent blocks is small. In [77], only the DC (zero frequency) and the lowest five AC (nonzero frequency) coefficients of a damaged block are estimated from the top and bottom neighboring blocks, while the rest of the AC coefficients are forced to be zeros. The DC values are linearly interpolated, and the five AC coefficients are synthesized according to the method specified in [78].

E. Recovery of Motion Vectors and Coding Modes

In the techniques described in Sections V-A to V-D, it is assumed that the coding mode and motion vectors are correctly received. If the coding mode and motion vectors are also damaged, they have to be estimated in order to use these methods for recovering lost coefficients. Based on the same assumption about spatial and temporal smoothness, the coding mode and motion vectors can be similarly interpolated from that of spatially and temporally adjacent blocks.

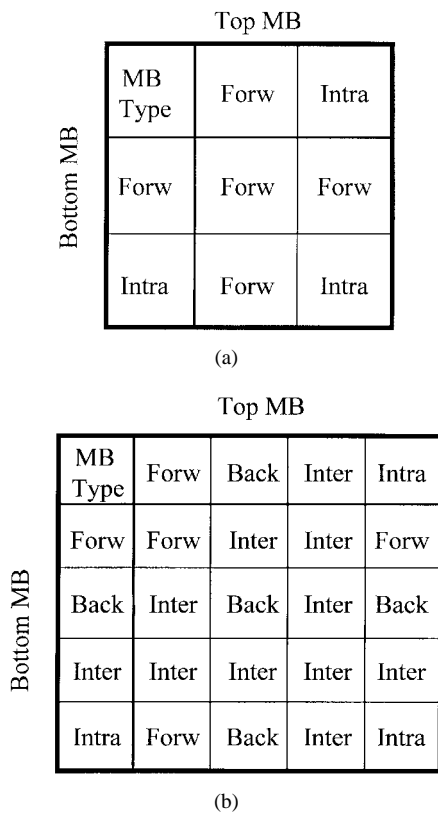


Fig. 10. Estimation of coding mode for MPEG-2. (a) P-frame. (b) B-frame. (Adapted from [77].)

For estimation of coding modes, the reconstruction scheme in [19] simply treats a block with a damaged coding mode as an intracoded block and recovers the block using information from spatially adjacent undamaged blocks only. This is to prevent any catastrophic effect when a wrong coding mode is used for such cases as scene change. Fig. 10 shows a more sophisticated scheme of estimating the macroblock coding mode from those of its top and bottom neighboring macroblocks for MPEG-2 coded video [77].

For estimation of lost motion vectors, the following methods have been proposed:

- a) simply setting the motion vectors to zeros, which works well for video sequences with relatively small motion;
- b) using the motion vectors of the corresponding block in the previous frame;
- c) using the average of the motion vectors from spatially adjacent blocks;
- d) using the median of motion vectors from the spatially adjacent blocks [79].

Typically, when a macroblock is damaged, its horizontally adjacent macroblocks are also damaged, and hence the average or median is taken over the motion vectors above and below. It has been found that the last method produces the best reconstruction results [79], [80]. The method in [81] goes one step further. It selects among

essentially the above four methods, depending on which one yields the least boundary matching error. This error is defined as the sum of the variations along the one-pixel-wide boundary between the recovered macroblock and the one above it, to its left, and below it, respectively. It is assumed that these neighboring macroblocks have been reconstructed previously, and for the damaged macroblock, only the motion vector is missing. In the event that the prediction error for this macroblock is also lost, then for each candidate motion vector, the boundary matching error is calculated by assuming that the prediction error of the damaged macroblock is the same as the top macroblock, the left one, the one below, or zero. The combination of the motion vector and the prediction error that yields the smallest boundary matching error is the final estimation solution. It was shown that this method yields better visual reconstruction quality than all of the previous four methods.

F. Summary

All the error-concealment techniques recover the lost information by making use of some *a priori* knowledge about the image/video signals, primarily the temporal and spatial smoothness property. The maximally smooth recovery technique enforces the smoothness constraint by minimizing the roughness of the reconstructed signal. The POCS method, on the other hand, iteratively projects the reconstructed image block onto the convex sets determined by the received coefficients and the smoothness constraint determined from the estimated edge direction of the block. Although generally giving more accurate results than the optimization approach, the POCS method is computationally more intensive, as it requires many iterations and estimation of edge directions. The interpolation method can be considered as a special case of the energy minimization approach when only the spatial difference measure is minimized. A problem with the direct spatial-domain interpolation approach is that it ignores the received coefficients in a damaged block. Both the POCS and the interpolation approaches only make use of spatial correlation. On the other hand, with the energy minimization framework used in the maximally smooth recovery method, both the spatial and temporal correlation can be exploited easily. The pros and cons of different methods are summarized in Table 3.

The reconstruction methods reviewed here are for transform-based coders using nonoverlapping transforms. Error-concealment techniques have also been developed for other coding methods, including subband [82]–[84], LOT [85], [86], and Walsh transform [87]. Fuzzy logic has also been used to recover high-frequency components that cannot normally be recovered by the smoothing and interpolation methods presented in this section [88]. Last, besides performing error concealment in the source-coder domain, as done by the techniques presented in this section, it is also possible to use residual redundancy from the source coder in the channel coder for error concealment. Since the output symbols of any source coder are not completely uncorrelated, this intersymbol correlation can

Table 3 Summary of Postprocessing Techniques for Error Concealment

| |
|---|
| Motion compensated temporal prediction: |
| Simple and gives good results if motion vectors of damaged blocks are available. Results are less reliable with estimated motion vectors. |
| Spatial interpolation [75,76]: |
| Simple, but can cause over blurring. Require block interleaving at the source end for better quality. Cannot make use of temporal correlation. |
| Maximally smooth recovery [19,71]: |
| Intermediate complexity, generally give good results, can deal with any loss patterns and exploit both spatial and temporal domain correlation. |
| POCS [74]: |
| Complex, can retain edge sharpness, quality depending on robustness of edge detection algorithms. Does not make use of temporal correlation. |
| Fuzzy logic [88]: |
| Complex, can preserve edge sharpness. |
| Recovery of coding mode and motion information [79,- 81]: |
| Simple. Erroneous estimate can lead to noticeable reconstruction errors. |

be used to improve the performance of the channel decoder in the presence of transmission errors. In [89], Sayood and Borkenhagen proposed to use a Viterbi decoder in front of the source decoder for this purpose.

VI. ENCODER AND DECODER INTERACTIVE ERROR CONCEALMENT

In the previous two sections, we described various techniques for error concealment from either the encoder or the decoder side, with little interaction between the two. Conceivably, if a backward channel from the decoder to the encoder is available, better performance can be achieved if the encoder and decoder cooperate in the process of error concealment. This cooperation can be realized at either the source coding or transport level. At the source coder, coding parameters can be adapted based on the feedback information from the decoder. At the transport level, the feedback information can be employed to change the percentage of the total bandwidth used for FEC or retransmission. In this section, we first describe several techniques that adapt the source-coding strategy based on the feedback information from the decoder. We then present a few schemes that vary transport level control. Retransmission, when used together with a conventional decoder, leads to decoding delays that may be unacceptable for real-time applications. Two novel schemes that counter this problem are described next. The first approach avoids the decoding delay by remembering the trace of damaged blocks at the decoder. The second scheme sends multiple copies of the lost data in each retransmission trial to reduce the number of retransmissions required. Although this technique can be applied in various video applications, we focus on its application in Internet video streaming. These two approaches are presented in more detail than other methods reviewed in this paper because they have not appeared in journal publications.

A. Selective Encoding for Error Concealment

The error-concealment problem would not be such an important issue for most real-time video-transmission applications if the encoder did not use prediction and therefore a bit error or packet loss did not cause error propagation. If errors only persist for one or two frames, the human eyes can hardly perceive the effect because it is too short. Temporal prediction is an indispensable building block in any video coder, however, because there is tremendous redundancy between adjacent video frames. Therefore, if the decoder can provide information about the locations of damaged parts to the encoder, the encoder can treat these areas differently so that the effect of error propagation can be either reduced or eliminated. One simple technique along this direction is that whenever the decoder detects an error, it sends a request to the encoder so that the next video frame is coded in intramode. This way, the error propagation will be stopped in about one round-trip time. Intracoding typically will reduce the compression gain, however, and hence degrade the video quality under the same bit-rate budget.

To reduce the bit-rate increase caused by intracoding, only part of the image needs intracoding due to the limited motion vector range [90], [91]. To further improve the coding efficiency, Wada proposed two schemes to perform selective recovery using error concealment [92]. When a packet loss is detected, the decoder sends the identity information of damaged blocks to the encoder. At the same time, error concealment is performed on the damaged blocks. Then normal decoding continues at the decoder. At the encoder side, two methods are proposed to stop error propagation at the decoder. In the first method, the affected picture area is calculated from the point of damaged blocks up to the currently encoded frame, as shown in Fig. 11(a). Then encoding is continued without using the affected area for prediction. Note that encoding without using the

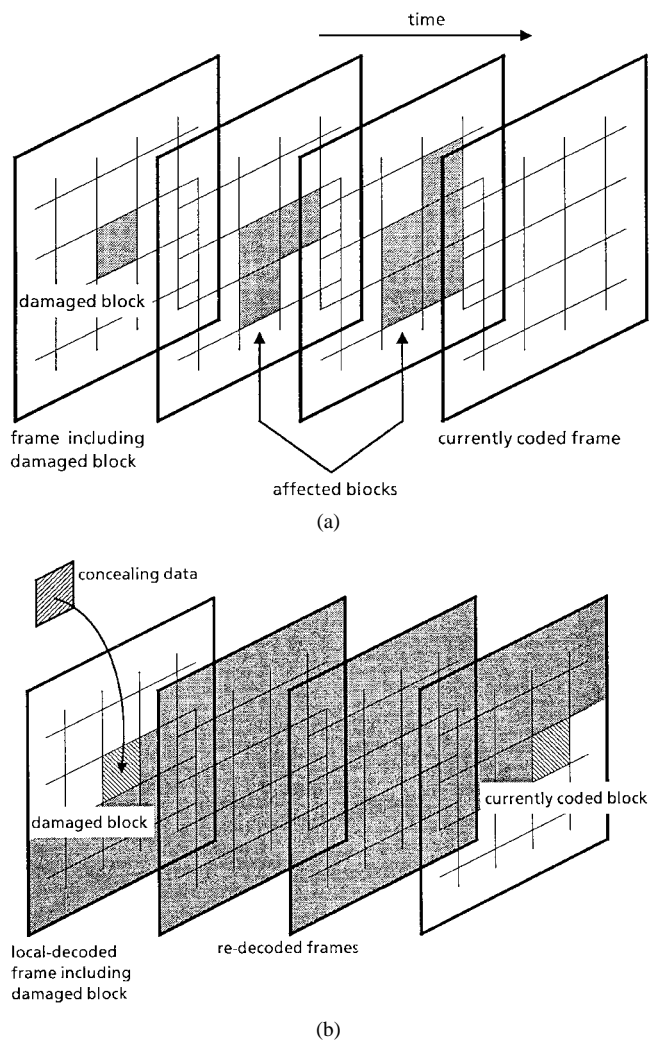


Fig. 11. Selective error concealment proposed by Wada. (a) Method A. (b) Method B. (From [92].)

affected area does not necessarily mean intracoding. In the second method, shown in Fig. 11(b), the same error-concealment procedure as that performed at the decoder is also carried out for the damaged blocks at the encoder. Then a local decoding is reexecuted from the point of the concealed blocks up to the currently encoded blocks. This is accomplished by using the transmitted data stored in the encoder transmission buffer so that the encoder's prediction frame buffer matches that at the decoder.

Similar to the above method, the H.263 standard [2] defines a *reference picture selection mode*, which is aimed at providing error resilience. In this method, both the encoder and the decoder have multiple prediction frame buffers. Fig. 12 shows a block diagram of such an encoder. Besides video data, the encoder and decoder exchange messages about what is correctly received and what is not. From this information, the encoder determines which frame buffers have been damaged at the decoder. Then the encoder will use an undamaged frame buffer for prediction. The information for the selected prediction frame buffer is also included in the encoded bit stream so that the decoder can use the same frame buffer for prediction.

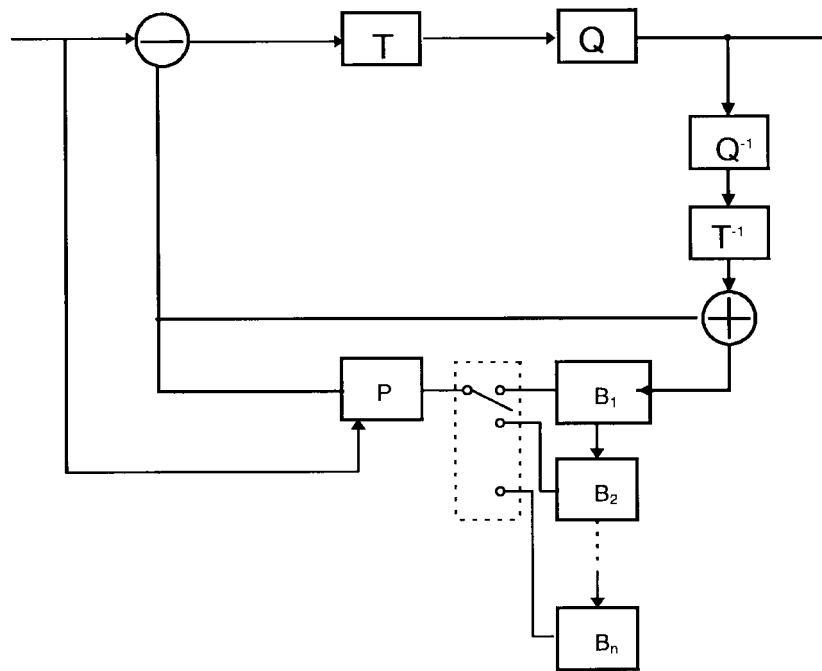
Horne and Reibman [93] proposed to send observed cell-loss statistics back to the encoder, which then adapts its coding parameters to match the prevailing channel conditions. More intrablocks and shorter slices are used when the loss rate is high, for enhanced error resilience, while fewer intrablocks and longer slices are invoked when the error rate is low, for improved compression efficiency.

B. Adaptive Transport for Error Concealment

In the last section, several techniques were described that adapt the source-coding strategy at the encoder based on feedback information from the decoder. In this section, we present several schemes that employ the feedback information for adjusting transport-level decisions. First, the transport controller can negotiate with the destination the retransmission of critical information that is lost. Retransmission has been used very successfully for nonreal-time data transmission, but it has been generally considered as unacceptable for real-time video applications because of the delay incurred. However, this viewpoint has changed slightly in the last few years. It has been realized that even for a coast-to-coast interactive service, one retransmission adds only about 70 ms of delay, which can be acceptable [94]. For one-way real-time video applications such as Internet video streaming and broadcast, the delay allowance can be further relaxed to a few seconds so that several retransmissions are possible. Retransmission has also been considered inappropriate for multipoint video conferencing because the retransmission requests from a large number of decoders can overwhelm the encoder. However, when a multipoint control unit (MCU) is used in a multipoint conference, the paths between the encoder and the MCU, and between the MCU and the decoders, are simply point to point. Retransmission can be applied in these paths separately. Another concern about using retransmission is that it may worsen the problem because it will add more traffic on the network and thus further increase the packet-loss rate. However, if retransmission is controlled appropriately, the end-to-end quality can be improved. For example, the encoder can reduce its current output rate so that the sum of the encoder output and the retransmitted data is kept below a given total data rate.

In spite of the above considerations, retransmission has not been used in most video-communications systems. This is mainly because most video applications are currently carried over ISDN networks, where the transmission error rate is relatively low. The error-propagation problem is circumvented by coding an entire frame in the intramode at the encoder when it is informed of the occurrence of a transmission error by the decoder. Recently, there has been an increasing interest in video communications over very lossy networks such as the Internet and wireless networks, and retransmission is expected to be deployed under these environments. In fact, both H.323 and H.324 standards have defined mechanisms of using retransmission for combating transmission errors [7], [9].

Because of its ubiquity, the Internet has been envisioned as the future platform for carrying various digital video



T Transform
 Q Quantizer
 P Motion compensated predictor
 B Frame buffer

Fig. 12. The reference picture selection mode of H.263.

services. However, the current Internet is a packet-based network with a best-effort delivery service. There is no end-to-end guaranteed QoS. Packets may be discarded due to buffer overflow at intermediate network nodes such as switches or routers or considered as lost due to excessive long queuing delay. Without any retransmission, experiments show that the packet loss rate is in the range of 2–10%, while the round-trip delay is about 50–100 ms on an average and can be more than 2 s in coast-to-coast connections [95]. With such error and delay characteristics, the achievable QoS is usually poor. Marasli *et al.* proposed to achieve better service quality in terms of delay and loss rate by using retransmission over an unreliable network [96]. Instead of trying retransmission indefinitely to recover a lost packet, as in TCP, the number of retransmission trials is determined by the desired delay. Smith proposed a cyclical user datagram protocol, which places the base-layer packets of a layered coder in the front of the transmission queue to increase the number of retransmission trials for the base layer [97]. Cen *et al.* and Chen *et al.* proposed to reduce the video output rate at the encoder when the network is congested [98], [99]. The feedback information about the network condition can be obtained by using the delay and loss-rate statistics at the decoder.

C. Retransmission Without Waiting

To make use of the retransmitted data, a typical implementation of the decoder will have to wait for the arrival

of the requested retransmission data before processing subsequently received data. This will not only freeze the displayed video momentarily, but also introduce a certain form of delay. If the decoder chose to decode faster than its normal speed after the arrival of retransmission, than only a few video frames would be displayed later than its intended display time; this delay is known as transit delay. On the other hand, the decoder can decode and display all the subsequent frames with a fixed delay, called accumulation delay. In the following, we describe a scheme that uses retransmission for recovering lost information for predictive video coders but does not introduce the delay normally associated with retransmission [100]. A similar technique was developed by Ghanbari, which is motivated by the desire of making use of late cells in an ATM network [101].

With this scheme, when a video data unit is damaged, a retransmission request is sent to the encoder for recovering the damaged data. Instead of waiting for the arrival of retransmitted data, the damaged video part is concealed by a chosen error-concealment method. Then normal decoding is continued, while a trace of the affected pixels and their associated coding information (coding mode and motion vectors) is recorded. The affected pixels refer to those that are subject to error-propagation effect of the damaged blocks. Upon the arrival of the retransmitted data, the affected pixels are corrected, so that they are reproduced as if no transmission loss had occurred. The correction

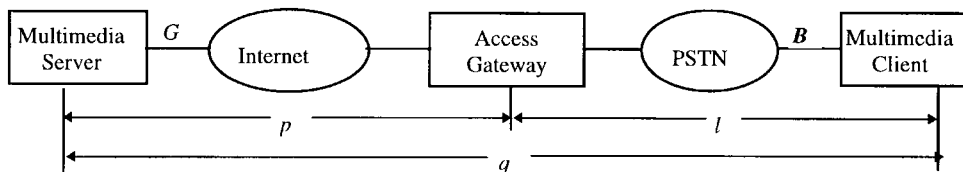


Fig. 13. Illustration of an Internet video-streaming application using a PSTN dial-up modem.

signal is obtained from the transmitted data and the recorded trace. When motion compensation is conducted in fractional pixel accuracy, tracing the affected pixels and generating the correction signal involves quite complicated processing. However, if only integer pixel motion compensation is used in the codec (for example, H.261 without the loop filter), then the above operations can be simplified greatly. In the following, we briefly describe the algorithm for this special case. The more general case can be found in [100]

Assume that a transmission error occurs at frame r , and the retransmitted data arrive at frame $r + d$.⁷ If the motion vectors are given in forward directions, then tracing the trajectory of a pixel is equivalent to summing the motion vectors along its route from the start frame r to the end frame $r + d$. However, motion vectors are described in the backward direction in all practical video codecs. Therefore, motion accumulation also has to be conducted in a backward direction. Let $p(i, m)$ represent the accumulated backward motion from frame i to frame r for the m th pixel in frame i and $v(i, m)$ represent the motion vector for this pixel from frame i to frame $i - 1$. Then $p(i, m)$ can be generated recursively using

$$p(i, m) = p(i-1, m) + v(i, m), \quad i = r+1, r+2, \dots, r+d.$$

To implement the above recursion, a frame buffer needs to be created to store the accumulated motions for all the pixels. The frame buffer is initialized to zero for the damaged pixels and to a special symbol for the undamaged pixels. The above recursion stops when a pixel falls in an intracoded block, which will also be signaled by the special symbol. Upon the arrival of the retransmission information, which includes the prediction errors $y(r, m)$ and motion vectors $v(r, m)$ for the damaged pixels in frame r , the correction signal at frame $r + d$ for the m th pixel is determined by

$$c(r+d, m) = y(r, m + p(r+d, m)) + z(r-1, m + v(r, m)) - \hat{z}(r, m)$$

where $z(r-1, m)$ is the reconstructed signal for pixel m at frame $r-1$ and $\hat{z}(r, m)$ is the error-concealment signal used at frame r for pixel m before receiving the retransmission data. If a pixel is assigned the special symbol in the frame buffer at frame $r+d$, it implies that this pixel is not affected by the transmission loss, and the above correction is not needed.

The above method can achieve lossless recovery except during the time between the information loss and the

⁷For simplicity, we assume that no other transmission errors occur during this time.

arrival of the retransmission data. During that interval, any postprocessing technique for error concealment described in the last section can be applied to the damaged regions. This scheme eliminates the delay associated with conventional retransmission schemes without compromising the video quality. The price paid is the relative high implementation complexity. However, when motion compensation is only applied at the integer pixel level, the implementation cost should be acceptable for most practical systems, as demonstrated above.

D. Prioritized, Multicopy Retransmission with Application to Internet Video Streaming

Given the stringent delay requirement for real-time video transmission, the number of admissible retransmissions is limited, and consequently, the residual error rate after the retransmission can still be high over very lossy networks such as the Internet or wireless networks. One way to reduce the residual error rate is by sending multiple copies of a lost packet in each single retransmission trial. With a network loss rate of l , the residual error rate can be reduced to $l^{1+\sum_{i=1}^L M_i}$ if L is the number of retransmission trials and M_i is the number of copies used for retransmission for the i th trial. For example, if $l = 0.1$, the residual error rate is reduced to 10^{-5} if $L = 2$ and $M_1 = M_2 = 2$. But in order to keep the overall output rate from the encoder under a given budget, the output rate from the source coder has to be reduced to accommodate the retransmission traffic. This can be accomplished by using layered coding at the source coder. When the network loss rate increases, the enhancement layers are partially transmitted or omitted entirely. For a lost packet, the number of retransmission trials and the number of retransmission copies are proportional to the importance of the layer to which the packet belongs. In the following, we demonstrate how to exploit this scheme to combat packet loss for an Internet video-streaming application.

Consider an Internet video-streaming application using the configuration shown in Fig. 13. The multimedia server sits on the Internet, and the client accesses files stored on the server through a dial-up modem link via the public switched telephone network (PSTN). Instead of downloading the entire file first and then playing it back, the file is played out as it is downloaded after a few seconds of initial delay. In this configuration, there are two main factors that affect the video quality at the client side. The first is the packet loss and delay jitter introduced in the Internet. The second is the relatively low channel capacity on the low-speed access link

Table 4 Summary of Interactive Error-Concealment Techniques

| |
|---|
| Selective encoding [2,92]: |
| Small overhead. Can be very effective when combined with restricted prediction coding. |
| Retransmission without waiting [100,101]: |
| Can achieve lossless recovery without the associated delay. High complexity. Less complex when integer motion vectors are used. |
| Prioritized multicopy retransmission: |
| Provide flexible tradeoff between delay and reconstruction quality. Can be effective for very lossy channels. |

from the client to the access gateway. In general, the loss rate in the PSTN is much lower than that in the Internet.

To cope with the high loss rate in the network, layered coding and multicopy retransmission are used. For a lost packet, more than one copy of retransmission is applied to both increase the probability of successful retransmission and reduce the number of retransmission trials, thus reducing delay. The number of retransmission copies for each layer is dependent on its importance to the reconstructed video quality. To avoid packet discarding at the access gateway, traffic arriving at the gateway cannot be greater than the access link capacity. In addition, the combined data output from the server (which consists of the streaming data and retransmission data) should be smaller than a certain value so as not to jam the Internet.

Let M be the number of layers used in coding the original multimedia data, where $M \geq 2$, and let R_i be the original data rate for layer i . Further, let C_{ij} represent the number of retransmission copies for layer i in the j th retransmission attempt and L the maximum retransmission trials allowed. Because of retransmission, the streaming rate for each layer R'_i , which is defined as the data rate excluding the retransmission data, may be different from the original coded data rate R_i . Hence, the combined data output from the server has to satisfy the following relationship:

$$\sum_{i=1}^M R'_i + \sum_{i=1}^M \left(\sum_{j=1}^L q^j C_{ij} \right) R_i = \min[G, B/(1-p)]$$

where q is the end-to-end packet-loss rate, G is the maximum data rate allowed by the server, B is the channel capacity of the access link, and p is the packet-loss rate in the Internet. In the above equation, the left side represents the combined traffic output from the server, with the first term accounting for the streaming rate and the second term for the retransmission rate. In general, p is unknown at both the server and the client. But q can be measured at either the client or the server, and the packet-loss rate in the access link l can also be obtained from the underlying physical layer or link layer. Then p can be derived according to $p = (q - l)/(1 - l)$.

To obtain the best video quality, the parameters L , R'_i , and C_{ij} have to be chosen jointly. The maximum number of retransmissions L is typically determined by the acceptable initial playout delay and the round-trip delay. For the more

important base layers, C_{ij} should be bigger than that for the enhancement layers. In addition, as the number of retransmission trials increases, and hence the remaining retransmission window narrows down, C_{ij} should increase for the base layer and decrease for the enhancement layers to yield more bandwidth for the base layer. For example, assume $M = 3$, $q = 0.1$, $l = 0.001$, $R_1 = 5$ kb/s, $R_2 = 10$ kb/s, $R_3 = 10$ kb/s, $L = 3$, $B = 25$ kb/s, and $G = 30$ kb/s. With C_{ij} chosen as $\{\{2, 3, 5\}, \{2, 1, 1\}, \{1, 1, 0\}\}$, we can achieve streaming rates of $R'_1 = 5$ kb/s, $R'_2 = 10$ kb/s, and $R'_3 = 7.4$ kb/s, with the residual error rates for the three layers being 10^{-11} , 10^{-5} , and 10^{-3} , respectively.

Because of the characteristics of the current Internet, sophisticated transport mechanisms such as the one presented in this section have to be applied to obtain an acceptable service quality. The permissible delay allowed by the video-streaming application makes it possible to use several retransmission trials so that a very low residual error rate can be achieved for the base layer. Like the scheme presented in the previous section, the improvement in the service quality is accomplished at the expense of an added implementation complexity. The sever and client have to not only support the prioritized multicopy retransmission protocol but also implement the layered coder structure.

E. Summary

In this section, we reviewed several techniques in the area of interactive error concealment. For applications that have a backward channel available from the decoder to the encoder, this class of methods should give the best performance because the redundancy is added only when an error occurs. This is especially important for channels that have bursty error characteristics. Table 4 summarizes the techniques we presented in this section. Note that these techniques can be used in conjunction with methods in the other two categories. In fact, the prioritized multicopy retransmission scheme is a combination of retransmission and layered coding.

VII. CONCLUDING REMARKS

In this paper, various techniques have been described for performing error concealment in real-time video communication. Depending on channel error characteristics and system configuration and requirements, some techniques are more effective than others. The burstiness of transmission

errors has a significant impact on the choice of algorithms. For a channel with bursty errors, forward error-concealment techniques may not be appropriate. This is because the overhead introduced by forward error concealment is wasted when the channel is error free, and such overhead is not very helpful when a burst error occurs. Retransmission may be more suitable since it only introduces the overhead when needed. The existence of a backward channel from the decoder to the encoder also affects the deployment of some schemes. In applications such as broadcast, where there is no backward channel, none of the interactive error-concealment techniques can be applied. The postprocessing techniques can be applied in any circumstances. However, the effectiveness of such techniques is limited by the available information. Also, some techniques may be either too complicated for cost-effective implementation or introduce unacceptable processing delay for real-time applications. Aside from the delay and complexity issues, one important criterion for comparing different schemes is the required concealment redundancy in source and/or channel coders to achieve the same degree of error protection. A fair comparison is difficult to obtain, however, because these techniques are usually developed for very different transport environments.

For future research, although more effective error-concealment approaches are still called for, more emphasis should be placed at the system-level design and optimization where the encoding algorithm, transport protocol, and postprocessing method should be designed jointly to minimize the combined distortion due to both compression and transmission. In addition, an optimal system should adapt its source-coding algorithm and transport-control mechanism to the network conditions so that the best end-to-end service quality is achieved. For example, a recently established transport protocol for mobile multimedia communication can provide several levels of error-resilience performance [102]. The system can hop between difference levels adaptively based on the error characteristics of the channel. However, there is very little interaction between the source coder and transport layer in terms of error concealment. An optimal system should allocate the concealment redundancy between the source coder and transport layers adaptively based on the channel environment so as to optimize the reconstructed video quality for a given decoder error-concealment capability. This remains a challenging task for future research and standardization efforts.

ACKNOWLEDGMENT

The authors wish to thank Dr. A. Reibman, AT&T Labs-Research, for carefully reading through the manuscript and providing many useful suggestions and critiques, as well as for her generous help in generating the images in Fig. 1. The authors also appreciate the constructive comments and suggestions from the anonymous reviewers, which have greatly improved the presentation of the paper.

REFERENCES

- [1] "Video codec for audiovisual services at $p \times 64$ kbits," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.261, 1993.
- [2] "Video coding for low bitrate communication," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.263, 1998.
- [3] "Information technology—Generic coding of moving pictures and associated audio information—Part 2: Video," ISO/IEC DIS 13818-2, 1994.
- [4] "Frame structure for a 64 to 1920 kbit/s channel in audio-visual teleservices," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.221, 1993.
- [5] "Narrow-band visual telephone systems and terminal equipment," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.320, 1993.
- [6] "Multiplexing protocol for low bitrate multimedia communication," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.223, 1995.
- [7] "Terminal for low bitrate multimedia communication," International Telecommunications Union, Geneva, Switzerland, ITU-T Draft Recommendation H.324, Dec. 1997.
- [8] "Media stream packetization and synchronization on non-guaranteed quality of services LANs," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.225.0, 1996.
- [9] "Telephone systems and equipment for local area networks which provide a non-guaranteed quality of service," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.323, Visual 1996.
- [10] N. Ohta, *Packet Video: Modeling and Signal Processing*. Norwood, MA: Artech House, 1994.
- [11] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [12] K. N. Ngan and R. Stelle, "Enhancement of PCM and DPCM images corrupted by transmission errors," *IEEE Trans. Commun.*, vol. COM-30, pp. 257–265, Jan. 1982.
- [13] K. M. Rose and A. Heiman, "Enhancement of one-dimensional variable-length DPCM images corrupted by transmission errors," *IEEE Trans. Commun.*, vol. 37, pp. 373–379, Apr. 1989.
- [14] O. R. Mitchell and A. J. Tabatabai, "Channel error recovery for transform image coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1754–1762, Dec. 1981.
- [15] W.-M. Lam and A. Reibman, "An error concealment algorithm for images subject to channel errors," *IEEE Trans. Image Processing*, vol. 4, May 1995, pp. 533–542.
- [16] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.
- [17] S. Vembu, S. Verdu, and Y. Steinberg, "The source-channel separation theorem revisited," *IEEE Trans. Inform Theory*, vol. 41, pp. 44–54, Jan. 1995.
- [18] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 801–806, June 1989.
- [19] Q.-F. Zhu, Y. Wang, and L. Shaw, "Coding and cell loss recovery for DCT-based packet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 248–258, June 1993.
- [20] K. Ramchandran, A. Ortega, and K. M. Uz, "Multiresolution broadcast for digital HDTV using joint source channel coding," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 6–23, Jan. 1993.
- [21] Y.-Q. Zhang, Y. J. Liu, and R. L. Pickholtz, "Layered image transmission over cellular radio channels," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 786–796, Aug. 1994.
- [22] M. Khansari and M. Vetterli, "Layered transmission of signals over power-constrained wireless channels," in *Proc. ICIP-95*, Washington, DC, Oct. 1995, vol. III, pp. 380–383.
- [23] M. Kansari *et al.*, "Low bit rate video transmission over fading channels for wireless microcellular systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 1–11, Feb. 1996.
- [24] R. Aravind, M. R. Civanlar, and A. R. Reibman, "Packet loss resilience of MPEG-2 scalable video coding algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 426–435, Oct. 1996.
- [25] J. K. Wolf, A. Wyner, and J. Ziv, "Source coding for multiple descriptions," *Bell Syst. Tech. J.*, vol. 59, pp. 1417–1426, Oct. 1980.
- [26] L. Ozarow, "On a source coding problem with two channels and three receivers," *Bell Syst. Tech. J.*, vol. 59, p. 1921, Dec. 1980.

- [27] A. A. El-Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 851–857, Nov. 1982.
- [28] Y. Wang and D. Chung, "Non-hierarchical signal decomposition and maximally smooth reconstruction for wireless video transmission," in *Mobile Multimedia Communications*, D. Goodman and D. Raychaudhuri, Eds. New York: Plenum, to be published.
- [29] A. S. Tom, C. L. Yeh, and F. Chu, "Packet video for cell loss protection using deinterleaving and scrambling," in *Proc. ICASSP'91*, Toronto, Canada, Apr. 1991, pp. 2857–2850.
- [30] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Trans. Inform. Theory*, vol. 39, pp. 821–834, May 1993.
- [31] V. A. Vaishampayan and J. Domaszewicz, "Design of entropy constrained multiple description scalar quantizers," *IEEE Trans. Inform. Theory*, vol. 40, pp. 245–251, Jan. 1994.
- [32] V. A. Vaishampayan, "Application of multiple description codes to image and video transmission over lossy networks," in *Proc. 7th Int. Workshop Packet Video*, Brisbane, Australia, Mar. 1996, pp. 55–60.
- [33] J.-C. Batllo and V. A. Vaishampayan, "Asymptotic performance of multiple description transform codes," *IEEE Trans. Inform. Theory*, vol. 43, pp. 703–707, Mar. 1997.
- [34] Y. Wang, M. T. Orchard, and A. R. Reibman, "Multiple description image coding for noisy channels by pairing transform coefficients," in *Proc. 1997 IEEE 1st Workshop Multimedia Signal Processing*, Princeton, NJ, June 1997, pp. 419–424.
- [35] M. T. Orchard, Y. Wang, V. Vaishampayan, and A. R. Reibman, "Redundancy rate-distortion analysis of multiple description coding using pairwise correlating transforms," in *Proc. IEEE Int. Conf. Image Process (ICIP97)*, Santa Barbara, CA, Oct 1997, vol. I, pp. 608–611.
- [36] S. S. Hemami, "Reconstruction-optimized lapped orthogonal transforms for robust image transmission," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 168–181, Apr. 1996.
- [37] D. Chung and Y. Wang, "Multiple description image coding using signal decomposition and reconstruction based on lapped orthogonal transforms," unpublished.
- [38] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 553–559, Apr. 1989.
- [39] M. T. Orchard, Y. Wang, A. R. Reibman, and V. Vaishampayan, "Multiple description coding by pairing transform coefficients," unpublished.
- [40] J. J. Spilker, Jr., *Digital Communications By Satellite*. Englewood Cliffs, NJ: Prentice-Hall, 1977.
- [41] J. Kurtenbach and P. A. Wintz, "Quantizing for noisy channels," *IEEE Trans. Commun. Technol.*, vol. CT-17, pp. 291–302, Apr. 1969.
- [42] N. Farvardin and V. Vaishampayan, "Optimal quantizer design for noisy channels: an approach to combined source-channel coding," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 827–838, Nov. 1987.
- [43] J. W. Modestino and D. G. Daut, "Combined source-channel coding of images," *IEEE Trans. Commun.*, vol. COM-7, pp. 1644–1659, Nov. 1979.
- [44] J. W. Modestino, D. G. Daut, and A. L. Vickers, "Combined source-channel coding of images using the block cosine transform," *IEEE Trans. Commun.*, vol. COM-29, pp. 1261–1273, Sept. 1981.
- [45] V. A. Vaishampayan and N. Farvardin, "Optimal block cosine transform image coding for noisy channels," *IEEE Trans. Commun.*, vol. 38, pp. 327–336, Mar. 1990.
- [46] S. S. Hemami and R. M. Gray, "Image reconstruction using vector quantized linear interpolation," in *Proc. ICASSP'94*, Australia, May 1994, pp. V-629–632.
- [47] S. Wenger, "Video redundancy coding in H.263+," in *Proc. AVSPN*, Aberdeen, U.K., Sept. 1997.
- [48] J. C. Maxted and J. P. Robinson, "Error recovery for variable length codes," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 794–801, 1985.
- [49] T. J. Ferguson and J. H. Ranowitz, "Self-synchronizing Huffman codes," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 687–693, July 1984.
- [50] P. G. Neumann, "Self-synchronizing sequential coding with low redundancy," *Bell Syst. Tech. J.*, vol. 50, pp. 951–981, Mar. 1971.
- [51] S.-M. Lei and M.-T. Sun, "An entropy coding system for digital HDTV applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, pp. 147–154, Mar. 1991.
- [52] W.-M. Lam and A. R. Reibman, "Self-synchronizing variable-length codecs for image transmission," in *Proc. ICASSP'92*, San Francisco, CA, Mar. 1992, pp. III477–480.
- [53] R. Koenen, "Overview of the MPEG-4 standard," ISO/IEC JTC1/SC29/WG11 N1730, July 1997.
- [54] T. Ebrahimi, "MPEG-4 video verification model version 8.0," ISO/IEC JTC1/SC29/WG11 N1796, July 1997.
- [55] N. T. Cheng and N. G. Kingsbury, "The EREC: An efficient error resilient technique for encoding positional information on sparse data," *IEEE Trans. Commun.*, vol. 40, pp. 140–148, Jan. 1992.
- [56] D. W. Redmill and N. G. Kingsbury, "The EREC: An error resilient technique for coding variable-length blocks of data," *IEEE Trans. Image Processing*, vol. 5, pp. 565–574, Apr. 1996.
- [57] R. Swann and N. G. Kingsbury, "Transcoding of MPEG-II for enhanced resilience to transmission errors," in *Proc. ICIP96*, Lausanne, Switzerland, vol. 2, pp. 813–816, Nov. 1996.
- [58] T. Kawahara and S. Adachi, "Video transmission technology with effective error protection and tough synchronization for wireless channels," in *Proc. ICIP96*, Lausanne, Switzerland, Nov. 1996, pp. 101–104.
- [59] S. H. Lee, P. J. Lee, and R. Ansari, "Cell loss detection and recovery in variable rate video," in *Proc. 3rd Int. Workshop Packet Video*, Morriston, Mar. 1990.
- [60] K. Challapali, X. Lebeque, J. S. Lim, W. H. Paik, R. Saint Girons, E. Petajan, V. Sathe, P. A. Snopko, and J. Zdepski, "The grand alliance system for US HDTV," *Proc. IEEE*, vol. 83, pp. 158–174, Feb. 1995.
- [61] T. Kinoshita, T. Nakahashi, and M. Maruyama, "Variable bit rate HDTV CODEC with ATM cell loss compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 230–237, June 1993.
- [62] V. Parthasarathy, J. W. Modestino, and K. S. Vastola, "Design of a transport coding scheme for high quality video over ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 358–376, Apr. 1997.
- [63] J.-Y. Cochenne, "Method for the correction of cell losses for low bit-rate signals transport with the AAL type 1," ITU-T SG15 Doc. AVC-538, July 1993.
- [64] E. Ayanoglu, R. Pancha, and A. R. Reibman, and S. Talwar, "Forward error control for MPEG-2 video transport in a wireless ATM LAN," *ACM/Baltzer Mobile Networks Applicat.*, vol. 1, no. 3, pp. 245–258, Dec. 1996.
- [65] T. Turelletti and C. Huitema, "RTP payload format for H.261 video streams," IETF RFC 2032, Oct. 1996.
- [66] C. Zhu, "RTP payload format for H.263 video streams," IETF draft, Mar. 1997.
- [67] H. Sun and J. Zdepski, "Error concealment strategy for picture-header loss in MPEG compressed video," in *Proc. SPIE Conf. High-Speed Networking and Multimedia Computing*, San Jose, CA, Feb. 1994, pp. 145–152.
- [68] M. R. Civanlar and G. L. Cash, "A practical system for MPEG-2 based video-on-demand over ATM packet networks," *Signal Process.*, vol. 8, pp. 221–227, Apr. 1996.
- [69] M. Ghanbari, "Cell-loss concealment in ATM video codes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 238–247, June 1993.
- [70] L. H. Kieu and K. N. Ngan, "Cell-loss concealment techniques for layered video codecs in an ATM network," *IEEE Trans. Image Processing*, vol. 3, pp. 666–677, Sept. 1994.
- [71] Y. Wang, Q.-F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. Commun.*, vol. 41, pp. 1544–1551, Oct. 1993.
- [72] W. Kwok and H. Sun, "Multi-directional interpolation for spatial error concealment," *IEEE Trans. Consumer Electron.*, vol. 39, pp. 455–460, Aug. 1993.
- [73] W. Zhu and Y. Wang, "A comparison of smoothness measures for error concealment in transform coding," in *Proc. SPIE Conf. Visual Communication and Image Processing*, Taipei, Taiwan, 1995, vol. II, pp. 1205–1214.
- [74] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Trans. Image Processing*, vol. 4, pp. 470–477, Apr. 1995.
- [75] S. S. Hemami and T. H.-Y. Meng, "Transform coded image reconstruction exploiting interblock correlation," *IEEE Trans.*

Image Processing, vol. 4, pp. 1023–1027, July 1995.

[76] S. Aign and K. Fazl, "Temporal & spatial error concealment techniques for hierarchical MPEG-2 video codec," in *Proc. Globecom'95*, pp. 1778–1783.

[77] H. Sun, K. Challapali, and J. Zdepski, "Error concealment in digital simulcast AD-HDTV decoder," *IEEE Trans. Consumer Electron.*, vol. 38, pp. 108–117, Aug. 1992.

[78] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand, 1992.

[79] P. Haskell and D. Messerschmitt, "Resynchronization of motion compensated video affected by ATM cell loss," in *Proc. ICASSP'92*, San Francisco, CA, Mar. 1992, pp. III545–548.

[80] A. Narula and J. S. Lim, "Error concealment techniques for an all-digital high-definition television system," in *Proc. SPIE Conf. Visual Communication Image Processing*, Cambridge, MA, 1993, pp. 304–315.

[81] W.-M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *Proc. ICASSP'93*, Minneapolis, Apr. 1993, pp. V417–V420.

[82] Y. Wang and V. Ramamoorthy, "Image reconstruction from spatial subband images and its applications in packet video transmission," *Signal Process.*, vol. 3, nos. 2–3, pp. 197–229, June 1991.

[83] S. S. Hemami and R. M. Gray, "Subband-coded image reconstruction for lossy packet networks," *IEEE Trans. Image Processing*, vol. 6, pp. 523–539, Apr. 1997.

[84] G. Gonzalez-Rosiles, S. D. Cabrera, and S. W. Wu, "Recovery with lost subband data in overcomplete image coding," in *Proc. ICASSP'95*, Detroit, MI, 1995, vol. II, pp. 1476–1479.

[85] K. H. Tzou, "Post filtering for cell loss concealment in packet video," in *Proc. SPIE Visual Communication Image Processing*, Philadelphia, PA, Nov. 1989, pp. 1620–1627.

[86] P. Haskell and D. Messerschmitt, "Reconstructing lost video data in a lapped orthogonal transform based coder," in *Proc. ICASSP'88*, Albuquerque, NM, Apr. 1990, pp. 1985–1988.

[87] W. C. Wong and R. Steele, "Partial correction of transmission errors in Walsh transform image coding without recourse to error correction coding," *Electron. Lett.*, May 1978, pp. 298–300.

[88] X. Lee, Y.-Q. Zhang, and A. Leon-Garcia, "Information loss recovery for block-based image coding techniques—A fuzzy logic approach," *IEEE Trans. Image Processing*, vol. 4, pp. 259–273, Mar. 1995.

[89] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Trans. Commun.*, vol. 39, pp. 838–846, June 1991.

[90] H. Yasuda, H. Kuroda, H. Kawanishi, F. Kanaya, and H. Hashimoto, "Transmitting 4 MHz TV signals by combinational difference coding," *IEEE Trans. Commun.*, vol. COM-25, pp. 508–516, May 1977.

[91] N. Mukawa, H. Kuroda, and T. Matuoka, "An interframe coding system for video teleconferencing signal transmission at a 1.5 Mbit/s rate," *IEEE Trans. Commun.*, vol. COM-32, pp. 280–287, Mar. 1984.

[92] W. Wada, "Selective recovery of video packet loss using error concealment," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 807–814, June 1989.

[93] C. Horne and A. R. Reibman, "Adaptation to cell loss in a 2-layer video codec for ATM networks," in *Proc. 1993 Picture Coding Symp.*, Mar. 1993.

[94] G. Ramamurthy and D. Raychaudhuri, "Performance of packet video with combined error recovery and concealment," in *Proc. IEEE Infocomm'95*, pp. 753–761.

[95] D. Sanghi, A. K. Agrawala, O. Gudmundsson, and B. N. Jain, "Experimental assessment of end-to-end behavior on Internet," in *Proc. IEEE Infocomm'93*, vol. 2, pp. 867–874.

[96] R. Marasli, P. D. Amer, and P. T. Conrad, "Retransmission-based partially reliable transport service: an analytic model," in *Proc. IEEE Infocomm'96*, San Francisco, CA, 1996, pp. 621–628.

[97] B. C. Smith, *Implementation Techniques for Continuous Media Systems and Applications*, Ph.D. dissertation, Univ. of California, Berkeley, 1994.

[98] S. Cen, C. Pu, R. Staehli, C. Cowan, and J. Walpole, "A distributed real-time MPEG video audio player," in *Proc. 5th Int. Workshop Network and Operating System Support for Digital Audio and Video*, Durham, NH, Apr 1995, pp. 151–162.

[99] Z. Chen, S.-M. Tan, R. H. Campbell, and Y. Li, "Real time video and audio in the world wide web," in *Proc. 4th World Wide Web Conf.*, 1995.

[100] Q.-F. Zhu, "Device and method of signal loss recovery for real-time and/or interactive communications," U.S. Patent 5 550 847, Aug. 1996.

[101] M. Ghanbari, "Postprocessing of late cells for packet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 669–678, Dec. 1996.

[102] "Multiplexing protocol for low bitrate mobile multimedia communications," International Telecommunications Union, Geneva, Switzerland, ITU-T Recommendation H.223/Annex A–Annex C, 1998.



Yao Wang (Member, IEEE) was born in Zhejiang, China, in 1962. She received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1983 and 1985, respectively, and the Ph.D. degree in electrical engineering from the University of California at Santa Barbara in 1990.

Since 1990, she has been a Member of the Faculty of Polytechnic University, Brooklyn, NY, where she currently is an Associate Professor of electrical engineering. From 1992 to 1996, she was a part-time consultant with AT&T Bell Laboratories, Holmdel, NJ. Since 1997, she has been a part-time consultant with AT&T Labs-Research, Red Bank, NJ. Since 1995, she has been an Associate Editor for the *Journal of Visual Communications and Image Representation*. She has been on the organizing/technical committees of several international conferences and workshops. Her current research interests include image and video compression for unreliable networks, motion estimation and object-oriented video coding, signal processing using multimodal information, and inverse scattering for medical imaging.

Dr. Wang is a member of the Technical Committee on Multimedia Signal Processing of the IEEE Signal Processing Society and the Technical Committee on Visual Signal Processing and Communications of the IEEE Circuits and Systems Society. Since 1995, she has been an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.



Qin-Fan Zhu (Member, IEEE) was born in Sichuan, China, in January 1964. He received the B.E. degree from Northwestern Polytechnic University, Xian, China, the M.E. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, and the M.S. and Ph.D. degrees from Polytechnic University, Brooklyn, NY, all in electrical engineering, in 1983, 1986, 1990, and 1993, respectively.

He was a Teaching and Research Fellow during 1983–1988 at UESTC and during 1990–1993 at Polytechnic University. From 1993 to 1996, he was with Motorola Information Systems Group, Mansfield, MA, as a Staff Engineer, where he was involved in research and development of various video-communications projects, including video transmission over asynchronous transfer mode networks, H.261/H.263-based software codecs, and Internet multimedia streaming. He participated in the International Telecommunications Union—Telecommunications Sector H.324 and H.323 standardization activities. In 1996, he joined MultiLink, Inc., Andover, MA, as a Principle Engineer, where he led the development of a continuous-presence video-processing system for an H.320-based multipoint control unit. Since 1997, he has been with PictureTel Corp., Andover, MA, which acquired Multilink. He currently is a Technical Manager leading the research, development, and architecture of digital video-processing systems for its server products. His research interests include video compression and transmission, multimedia system architecture, and implementation. He has received three patents with four more pending.